

# SOVELLUSKEHITYS RULLANKÄSITTELYAUTOMAATION MODERNISOINTIPROJEKTISSA

Jere Juka

Opinnäytetyö  
Huhtikuu 2011

Paperikoneteknologia  
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU  
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) JUKA, Jere	Julkaisun laji Opinnäytetyö	Päivämäärä 25.04.2011
	Sivumäärä 49+4	Julkaisun kieli suomi
	Luottamuksellisuus ( ) saakka	Verkojulkaisulupa myönnetty ( X )
Työn nimi SOVELLUSKEHITYS RULLANKÄSITTELYAUTOMAATION MODERNISOINTIPROJEKTISSA		
Koulutusohjelma Paperikoneteknologia		
Työn ohjaaja(t) STRÖM, Markku		
Toimeksiantaja(t) Protacon Oy, LEMINEN, Jyrki		
<p>Tiivistelmä</p> <p>Työn toimeksiantaja Protacon Oy oli solminut käynnistyväksi automaation modernisointiprojektin paperitehtaan kuljetinjärjestelmän ja rullarampiston ohjauslaitteiden uusimiseksi. Koska nykyisen järjestelmän laitteistotuki oli lakannut, sen luotettavuus oli muuttunut kyseenalaiseksi. Tavoitteena oli uuden järjestelmän sovelluskehitys ylläpitäen järjestelmän toiminnallisuus vanhaan verrattuna samana.</p> <p>Toteutus pohjautuu Siemens Simatic S7-teknologiaan, joka ei sovellusohjelmiensa kannalta ole yhteensopiva tehtaan vanhojen järjestelmien kanssa. Ei ollut myöskään olemassa valmiita työkaluja näiden ohjelmien kääntämiseksi, joten logiikkaohjelmat kirjoitettiin alusta asti uudelleen tutkien asiakkaan toimittamia ohjelmalistauksia. Uusien toimintojen osalta kehitystyö tehtiin asiakkaan tarpeita mukaillen.</p> <p>Työn loppuvaiheilla koottiin toimitussisällön päälogiikat hajautusyksiköineen sekä operointipaneeli koestusympäristöksi sovellusohjelmien tarkastusta varten. Tämä testausvaihe tarjosi arvokasta tietoa uuden laitteiston toiminnasta. Testin perusteella voidaan todeta ohjelmointityön ja sovelluskehityksen onnistuneen tavoitteiden mukaisesti. Vanhasta järjestelmästä periytyvät ohjelmarutiinit yltyivät toiminnaltaan toivotulle tasolle. Sovellusohjelmaa saatiin paikoin kehitettyä myös aiempaa tehokkaammaksikin. Työn lopullinen tulos selviää käyttöönotossa viimeisten viritysten jälkeen.</p> <p>Työn tulosten perusteella on turvallista todeta eri logiikkalaitteiden toiminnallisesti näennäisen samankaltaisuuden pitävän sisällään täysin erilaiset järjestelmärajapinnat. Nämä saattavat nousta haasteiksi ohjelmointityössä varsinkin sovellusohjelmia järjestelmästä toiseen sovitettaessa. Tämän työn tuloksia voidaan hyödyntää automaation modernisointien toteutustapaa pohdittaessa, kun valinta pitää tehdä konversion tai uuden sovelluksen kehityksen välillä.</p>		
Avainsanat (asiasanat)		
Allen-Bradley, PLC-2, Siemens, Simatic S7, logiikat, operointipaneeli, kuljettimet		
Muut tiedot		



Author(s) JUKA, Jere	Type of publication Bachelor's Thesis	Date 25042011
	Pages 49+4	Language Finnish
	Confidential ( ) Until	Permission for web publication ( X )
Title APPLICATION DEVELOPMENT FOR A ROLL HANDLING AUTOMATION MODERNIZATION PROJECT		
Degree Programme Paper Machine Technology		
Tutor(s) STRÖM, Markku		
Assigned by Protacon Oy LEMINEN, Jyrki		
<p>Abstract</p> <p>The client Protacon Oy had negotiated a contract of an automation modernization project for a paper mill's conveyor and roll ramp systems' control devices renewal. Since the current system hardware support had ceased, its reliability had become questionable. The aim was an application development for the new system to maintain the same functionality compared to the old one.</p> <p>Implementation is based on the Siemens Simatic S7 whose application programs are not compatible with the existing systems of the factory. Nor was there any ready-made tools for converting these programs, they were written from scratch by researching the old program listings supplied by the customer. The development of the new functions was made to adapt the customer's needs.</p> <p>In the final stages of the work the main logic controllers followed by their I/O distribution units and the operation panel were assembled together to create an environment for the factory acceptance testing. In this stage the developed programs were uploaded to the controllers for an inspection. The test phase provided valuable information about the functionality of the new equipment. Based on the test it can be said that the programming process and the application development succeeded as intended. The program routines inherited from the old system reached the desired level of function. Meanwhile some of the logic programs were also compiled so that they run even more efficiently. The final result of this work can be evaluated after the new system is tuned and taken in use.</p> <p>Based on the results it is safe to say that different logic devices' apparent similarity in functionality encompasses completely different system interfaces. These challenges may increase in the programming phase, especially where the application software is adjusted to another system. The results of this work can be utilized in automation modernization projects when the choice of execution between conversion and new application development is considered.</p>		
Keywords  Allen-Bradley, PLC-2, Siemens, Simatic S7, logics, operation panels, conveyors		
Miscellaneous		

# SISÄLTÖ

1 TEHTÄVÄ JA TAUSTATIEDOT .....	2
1.1 Yritysesittely .....	2
1.2 Tehtävän lähtökohdat, rajausta ja tavoite .....	5
1.3 Rullarampiston toimintaperiaate .....	6
2 OHJELMOITAVAT LOGIIKAT .....	8
2.1 Logiikkakomponenttien valinta .....	9
2.1.1 Tuloyksikkö .....	10
2.1.2 Lähtöyksiköt .....	12
2.1.3 Keskusyksikkö .....	13
2.2 Logiikan hajautus .....	14
2.3 Logiikan ohjelmointi .....	16
2.3.1 Ohjelman rakenne .....	18
2.3.2 Ohjelmointilaitteet .....	19
2.4 Operointipaneelit (HMI, Human Machine Interface) .....	21
3 OHJELMOINTIYMPÄRISTÖT .....	22
3.1 SIMATIC Step 7 Professional .....	23
3.2 WinCC flexible 2008 .....	24
4 OHJELMOINTI .....	25
4.1 Vanhan ohjelman konversio .....	25
4.2 Rulladatan siirto .....	28
4.3 Valvomo-ohjelmointi .....	33
4.3.1 Alakerran näytöt .....	35
4.3.2 Yläkerran näytöt .....	39
4.3.3 Hälytysnäytöt .....	43
5 KOESTUS JA KÄYTTÖÖNOTTO .....	44
6 POHDINTA .....	46
LÄHTEET .....	48
LIITTEET .....	49
Liite 1. Rulladatan siirto rullankäsittelylaitteilla .....	49
Liite 2. Logiikkalaitteiston väylärakenne, Siemens NetPro .....	50
Liite 3. Logiikkalaitteisto, pakkaamon etupuoliset kuljettimet .....	51
Liite 4. Logiikkalaitteisto, pakkaamon jälkeiset kuljettimet .....	52

## KUVIOT

Kuvio 1. Protacon Groupin visio, missio ja arvot (Protacon Group 2011) .....	2
Kuvio 2. Protacon Groupin henkilöstö ja liikevaihto (Protacon Group 2011).....	3
Kuvio 3. Protacon Group, liikevaihto toimialoittain (Protacon Group 2011) .....	4
Kuvio 4. Rullarampiston rakenne .....	6
Kuvio 5. Ohjelmoitavat logiikat nyt ja ennen .....	8
Kuvio 6. Binäärinen tuloyksikkö toimintakaavioineen, SM 321 (Siemens AG 2011) ...	11
Kuvio 7. Binäärinen lähtöyksikkö toimintakaavioineen, SM 322 (Siemens AG 2011) .	12
Kuvio 8. Keskusyksikkö I/O -kortein, CPU 312 (Siemens AG 2011) .....	13
Kuvio 9. Hajautuksen verkkotopologiat .....	15
Kuvio 10. Logiikkakaavio (FBD) esitysmuotona STEP 7-ohjelmassa.....	16
Kuvio 11. Relekaavio (LAD) esitysmuotona STEP 7-ohjelmassa.....	17
Kuvio 12. Käskylista (STL) esitysmuotona STEP 7-ohjelmassa .....	17
Kuvio 13. SIMATIC Manager .....	23
Kuvio 14. WinCC flexible .....	25
Kuvio 15. Ramppiohjauksien ohjelmakutsu .....	27
Kuvio 16. Ote ramppiohjauksien koodista ja parametreista .....	28
Kuvio 17. Rulladatansiirto kuljettimilla, rullanpakkauskoneen alue.....	29
Kuvio 18. Rulladatansiirto kuljettimilla, rampiston alue .....	30
Kuvio 19. Sanomanvälitys, PTS-kuljetinlogiikka .....	31
Kuvio 20. Rullan ramppiosoitteen kyselysanoma .....	32
Kuvio 21. Rullan ramppiosoitteesanoma.....	32
Kuvio 22. Operointipaneeli MP 377 12" key (Siemens AG 2011) .....	34
Kuvio 23. Alakerran layout ja rampisto .....	35
Kuvio 24. Ramppien rajakytkintiedot.....	36
Kuvio 25. Data rampeilla .....	37
Kuvio 26. Ramppidatan editointi.....	38
Kuvio 27. Yläkerran layout .....	39
Kuvio 28. Kuljettimilla oleva rullamuisti: FIFO -rekisterit.....	40
Kuvio 29. FIFO -rekisterin Help-ikkuna.....	41
Kuvio 30. Kuljettimilla oleva rullamuisti: FIFO -rekisterin resetointi .....	41
Kuvio 31. Kuljettimilla oleva rullamuisti: FIFO -rekisterin editointi .....	42
Kuvio 32. Hälytysnäyttö.....	43
Kuvio 33. Hälytyshistoria .....	44
Kuvio 34. FAT -kokoonpano .....	45

# 1 TEHTÄVÄ JA TAUSTATIEDOT

## 1.1 Yritysesittely

Protacon Group on vuonna 1990 Keski-Suomessa perustettu teknologia-alan suunnittelu- ja palveluyritys. Alkujaan teollisuuden automaatio-sovelluskehityksiin keskittynyt yhtiö on laajentanut toimialaansa vuosien varrella muun muassa yritysostoin. Nykyään konserni koostuu seitsemästä tulostavasta osakeyhtiöstä, jotka ovat emoyhtiö Protacon Oy, Protacon Solutions Oy, Protacon Software Oy, Sysdrone Oy, Protacon Logix Oy, AA-Control Oy ja Liqum Paper Oy. (Protacon Group 2011)

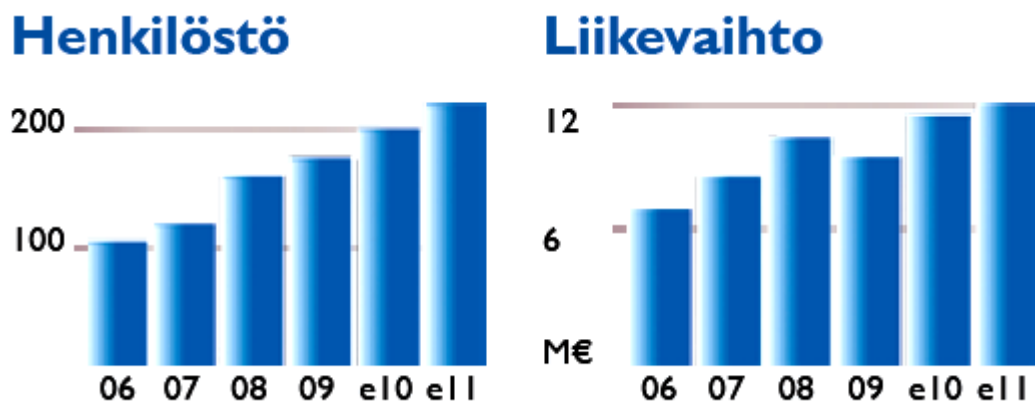
Kilpailukykyä moneen pienempään toimijaan verrattuna tuo se, että saman katon alta voidaan toimittaa kokonaisvaltaisesti konsultoinnin, suunnittelun, toteutuksen ja ylläpidon toiminnot liittyen automatisointiin, sähköistykseen, instrumentointiin sekä mittaus- ja informaatiojärjestelmiin. Protacon Groupin asiakkaisiin lukeutuvat energia-, metsä- ja konepajateollisuuden sekä infrastruktuurialojen yritykset. Tavoitteellisen toiminnan näkökulmasta koko konsernin liiketoimintaa ohjaavat kuviossa 1 esitetyt kulmakivet. (Protacon Group 2011)



Kuvio 1. Protacon Groupin visio, missio ja arvot (Protacon Group 2011)

## Protacon Group lukuina

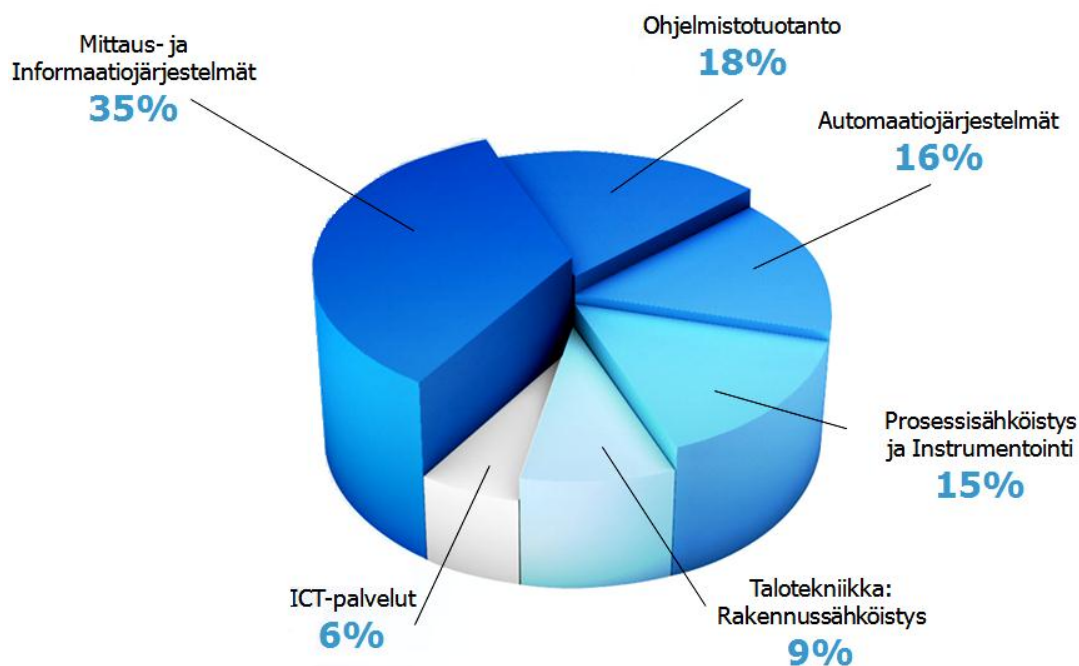
Konsernilla on alueyksiköitä yhdellätoista eri paikkakunnalla Suomessa. Pääkonttori sijaitsee Jyväskylässä ja muut toimipisteet ovat Helsingissä, Hollolassa, Kajaanissa, Kalajoella, Kotkassa, Oulussa, Savonlinnassa, Tampereella, Vaasassa ja Varkaudessa. Näiden toimipisteiden kirjoilla on yhteensä noin 180 automaation, sähköistyksen ja informaatioteknologian ammattilaista. Vuodelle 2009 liikevaihtoa kertyi 10milj. euroa. Liikevaihdon ja henkilöstömäärän kehitys ennusteineen on esitetty kuviossa 2. (Protacon Group 2011)



Kuvio 2. Protacon Groupin henkilöstö ja liikevaihto (Protacon Group 2011)

## Toiminta-alueet

Protacon Groupin toiminta-alueet voidaan jakaa kolmeen osioon. Ensimmäinen pitää sisällään teollisuuden suunnittelupalvelut sekä niihin liittyvät järjestelmätoimitukset osa-alueinaan: sellu ja paperi, mekaaninen puunjalostus, energia-, metalli-, ajoneuvo- ja muu teollisuus. Toinen suuri toiminta-alue on informaatioteknologian ohjelmistokehitys ja siihen liittyvät ICT-palvelut. Kolmas alue on talotekniikka, joka kattaa rakennussähköistyspalvelut ja kiinteistöautomaation. Edellisten keskinäisistä suuruusluokista antaa osviittaa kuviossa 3 esitetty liikevaihdon jakautuminen toimialoittain. (Protacon Group 2011)



Kuvio 3. Protaccon Group, liikevaihto toimialoittain (Protaccon Group 2011)

## Protaccon Oy

Edellä kerrotuista toiminta-alueista Protaccon Oy:n tuottamiin palveluihin kuuluvat automaation ja sähköistuksen ratkaisut sekä suunnittelu- ja asiantuntijapalvelut. Yhtiön toiminta ulottuu projektin alkuvaiheen tarvemäärityksistä investointilaskelmiin, näihin pohjautuvaan suunnitteluun ja edelleen valmiin projektin toimituksista aina käyttöönottoihin saakka. Toteutukset ulottuvat pienemmistä toimituksista vaativiin automaation kokonaistoimituksiin, ja suurten toimeksiantajien myötä valtaosa toiminnasta suuntautuu ulkomaille. Toiminnan johtamisesta ja yhtiön taloudellisesta tilasta kertoo jotain se, että lamasta huolimatta luottoluokituslaitos Dun & Bradstreet Finland Oy on luokitellut Protaccon Oy:n korkeimpaan luottoluokkaan AAA. (Protaccon Group 2011)

Automaatiosovellusten ja – järjestelmien alueelta Protaccon Oy:n henkilöstöltä löytyy asiantuntemusta ja suunnitteluosaamista muun muassa seuraavista järjestelmistä: MetsoDNA ja Damatic XD/XDi, Siemens S5, S7, PCS7 ja WinCC, Allen-Bradley PLC2-, PLC5-, SLC500- ja ControlLogix, ABB, Omron, National Instruments sekä lukuisat muut. Osaaminen ja kokemus helpottavat paitsi oikean järjestelmän tarjoamista ja



räätälöimistä kulloiseenkin toimitukseen, myös hyödyntämään valitun järjestelmän ominaisuudet tehokkaasti. (Protacon Group 2011)

## 1.2 Tehtävän lähtökohdat, raja- ja tavoite

Mahdollisuus tähän opinnäytetyöhön tarjoutui projektissa, jossa Protacon Oy toimittaa paperitehtaalle rullankäsittelyyn kytkeytyvän automaation modernisoinnin. Teutetus kattaa vaippakäärintäaseman, rullankuljetinjärjestelmän sekä rullarampiston automaatiolaitteistot. Uudistetun järjestelmän käyttöönotto tulee olemaan alkuvuodesta 2011.

Asiakkaan puolesta nyt käynnistyvälle modernisointiprojektille on ollut tarve jo 90-luvulta alkaen, sillä tehtaalla toistaiseksi käytössä olevat, sinällään vielä täysin toimintakuntoiset logiikkalaitteistot ovat kovasti vanhentuneet. Käytössä olevalle 80-luvulta periytyvälle Allen Bradley PLC2-laitteistolle ei löydy enää tuotetukea, eikä varaosia valmistajan toimesta. Lisäksi laitteiston ylläpidon ja ohjelmoinnin hallitsevien ammattilaisten rivit käyvät harvoiksi verrattuna nykypäivänä laajemmin käytössä oleviin prosessiautomaation laitteisiin.

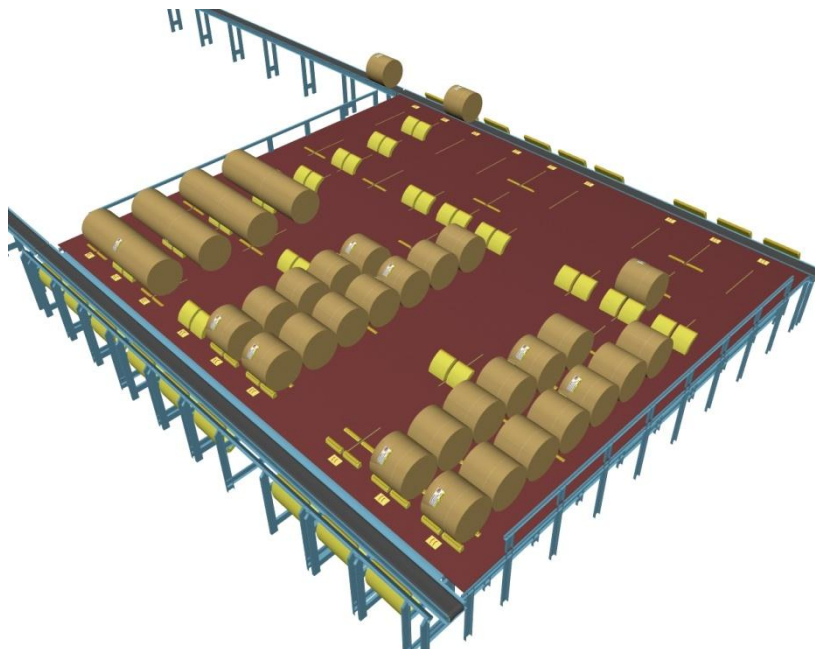
Modernisointisopimuksen toimitussisältöön kuuluu järjestelmä- ja sähkösuunnittelu, uudet Siemens Simatic S7-järjestelmään pohjautuvat logiikkalaitteistot ja operointipaneelit, edellisten ohjelmointityö, FAT -testaus ja käyttöönotto sekä käyttöhenkilöstön koulutus. Projektin lähtödokumenteiksi asiakas toimitti piiri- ja moottoriluettelot sekä johdotustaulukot ja kaapeliluettelot. Käytettäväksi oli toimitettu myös korvattavan järjestelmän logiikkaohjelmisto, joka osoittautui ohjelmointityön edetessä vanhentuneeksi. Tämän käytyä ilmi, onnistui asiakas kuitenkin toimittamaan logiikoissa toiminnassa olevan suorituksenaikaisen version ohjelmistosta.

Sinällään harmittoman oloisella seikalla oli kuitenkin työläs vaikutus, sillä PLC2- ja Simatic S7-järjestelmien sovellusten konvertointiin ei ole olemassa työkaluja, vaan käännöstyö ohjelmoidaan käsin alusta alkaen. Tämä huomioiden lienee oikeutettua luonnehtia projektia varsin mittavaksi, joten opinnäytetyön kannalta kokonaisuutta

oli rajattava. Tässä tapauksessa aihealueeksi valittiin rullarampiston sovelluskehitys, johon sisältyy logiikka- ja operointipaneelin ohjelmointi sekä rullatiedon käsittely pakkauskoneen jälkeisillä kuljettimilla. Työn tavoitteeksi asetettiin rullarampiston nykyisen toiminnallisuuden säilyttäminen sekä selkeän ja toivotut toiminnot omaavan valvomo-ohjelmiston kehittäminen operointipaneelia varten.

### 1.3 Rullarampiston toimintaperiaate

Tuotantolaitoksella konerullan tullessa pituusleikkurille, joudutaan se tavallisesti leikkaamaan useamman eri tilauksen asiakasrulliksi. Näin toimimalla pyritään minimoimaan leikkaus- eli trimmihävikki. Monissa tapauksissa rullat lajitellaan heti pituusleikkauksen jälkeen omiin lähetyseriinsä, sillä tämä tehostaa jäljempänä tapahtuvia varasto-operaatioita. Kun yhtäikaa toimivia pituusleikkureita on monta tai jos jälkikäsittelyalueella on useampia rullanpakkauskoneita, rullien käsittely tuotannonsuunnittelun eräohjeiden mukaisiin kokonaisuuksiin voi olla vaikeaa. Tällöin otetaan usein käyttöön rullarampisto. (KnowPap 2010)



Kuvio 4. Rullarampiston rakenne

Rullarampisto koostuu tulo- sekä purkukuljettimista ja yksittäisten ramppien rullan pukkaamista ja vastaanottimista. Rampiston rakenteen ymmärtämisessä voi helpottaa sen mieltäminen esimerkiksi kaksiulotteiseksi matriisiksi. Sarakkeiden lukumäärä vastaa ramppien määrää ja rivien lukumäärä vastaavasti yhdelle rampille peräkkäin mahtuvien rullien määrää. Mikäli ajatusleikkiä jatketaan edelleen kuljettimiin, huomataan että toinen ulottuvuudesta katoaa. (Leminen 2010)

Rullarampisto toimii tulo- ja purkukuljettimen välisenä lajittelupuskurina. Tulokuljettimelta satunnaisessa järjestyksessä saapuvat eri lähetyserien rullat ohjataan rampeille siten, että samaa tilausta olevat rullat päätyvät samalle rampille. Nyt yksittäisellä rampilla olevat rullat kuuluvat siis samaan asiakastilaukseen ja voidaan myös olettaa, että ne tullaan sijoittamaan tuotevarastossa samaan varastopaikkaan. Seuraavaksi rampilta puretaan rullien koosta riippuen 1-4 rullan erä peräkkäin purkukuljettimelle. Tavallisimmissa sovelluksissa tämä erä käännetään pystyynnostimella pyörivästä asennosta pystyasentoon. Tällainen pino on ryhmitelty valmiiksi, jolloin trukki tai muu kuljetin pystyy siirtämään useamman rullan yhtäaikaaisesti. Ryhmittely helpottaa myös lastausta, vähentäen rullankäsittelymäärää. Samalla pienenee paitsi lastausaika myös rullien kolhiintumisriski. (Leminen 2010)

Miten yksittäiset rullat sitten löytävät paikkansa rampistolta? Automaation näkökulmasta kyseessä on osoitejärjestelmä, jossa jokaisella rampilla on oma yksilöity osoitteensa. Rullatietojen ja siten myös osoitejärjestelmän hallinta on rampistoa ohjaavan logiikan yläpuolisen järjestelmän hallinnassa, jota ei käsitellä tarkemmin tässä työssä. Ylemmän tason järjestelmältä saatu osoitetieto kirjoitetaan esimerkiksi rullanpakkauskoneelta poistuttaessa rullaa kuljettimien lävitse seuraavaan datasanaan. Juuri tämän datan vertailulla rampiston laitteita ohjaava logiikka osaa lajitella rullat oikeille paikoilleen. Kuten ohjausalueita yleensä, on myös kaikkia rampiston toimintoja pystyttävä hallitsemaan käsiajotilassa esimerkiksi vaikeimmissa häiriötilanteissa. Siinä mielessä järjestelmän pitää kuitenkin olla jo itsessään häiriösietoinen, että kohderampin ollessa joko täynnä tai poissa käytöstä, koko järjestelmä ei seisaudu, vaan rullat voidaan purkaa pois jonkun toisen rampin kautta automaattijollakin. Sama koskee tilannetta osoitetiedon kadotessa esimerkiksi resetoinnin yhteydessä. Kuva-

uksen tapainen läpiajoramppi sijaitsee useimmiten tulokuljettimen ajosuunnassa viimeisenä. Silloin yhdellekään aikaisemmalle rampille kuulumaton rulla siirtyy viimeistään viimeiseltä rampilta poistokuljettimelle ja sitä kautta edelleen varastoitavaksi. Sovelluksen niin vaatiessa rampistosta voi löytyä muitakin erikoisramppeja, kuten pikaramppeja kiiretilauksille. Pikaramppien kautta rullat pääsevät heti läpi, kun ramppi on tyhjä. (Leminen 2010)

## 2 OHJELMOITAVAT LOGIIKAT

Ohjelmoitava logiikka (PLC, Programmable Logic Controller) kääntyisi englanninkielisestä termistään suoraan sanasta sanaan ohjelmoitavaksi logiikkasäätimeksi. Nykyaikaiset logiikkalaitteistot ovat kuitenkin toiminnoiltaan huomattavasti monipuolisempia ohjausjärjestelmiä verrattuna tavanomaiseen, asetusarvon ja mittauksen erotukseen perustuvaan säätimeen.



Kuvio 5. Ohjelmoitavat logiikat nyt ja ennen

Historia muistiin ohjelmoitavien logiikkalaitteiden taustalla vie 60-luvulle, ja ensimmäisen laitteen toi tuolloin markkinoille General Motors. Kun ymmärrettiin ohjelmoitavien laitteiden ohjaamien järjestelmien nopea ja joustava muuntautumiskyky vaikkapa tuotantolinjan muutoksessa, alkoivat laitteet syrjäyttää tavanomaista releohjausta nopeasti. Itse asiassa jo vuonna 1994 Frost & Sullivanin koostamassa markkinatutkimuksessa ennustettiin vuosituhannen taitteessa myytävän globaalisti liki 18 miljoonaa logiikkaa. (Parr 1999, 18-20; Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 102)

Valinnanvaraa laitteiden kirjosta sitten löytyykin, sillä jo logiikkamarkkinoiden valtanimien (Siemens, Allen Bradley, Omron jne.) omissa valikoimissa on runsaasti vaihtoehtoja niin tulojen ja lähtöjen laajuuden, eli I/O-määrän, kuin suorituskyvyn mukaan. Valmistajasta tai logiikan tyypistä riippumatta voidaan kuitenkin huomata laitteiden toiminnan olevan lähellä toisiaan, sillä tuoteperheen nimityksen mukaisesti ne ovat perustoiminnoiltaan kaikki ohjelmoitavissa loogisilla käskyillä. Tämä tarkoittaa sitä, että logiikkaan tulevat anturitiedot otetaan vastaan ja saatuun dataan reagoidaan käyttäjän muistiin määrittelemän sovellusohjelman mukaisesti. Ohjelmaa vastaava reaktio voi olla esimerkiksi toimilaitteen ohjaus, ja sen reaktionopeuden määrittää pääasiassa laitteiston suorituskky. Etunsa eri logiikkojen välisten toimintojen samankaltaisuudella on siinä mielessä, että yhden ohjelmoitavan logiikan perusteellinen hallinta auttaa myös jonkun muun valmistajan laitteen käytön oppimisessa. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 102)

Elektroniikan innovaatioiden mahdollistama nopea kehitys on tuonut ohjelmoitaviin logiikoihin jatkuvasti lisää toimintoja ja laitteiden suorituskky on ottanut aimo harppauksia eteenpäin prosessorien tehostumisen myötä. Nykyiset laajemmat logiikkajärjestelmät alkavatkin muistuttaa jo kokonaisia automaatiojärjestelmiä. Ero edellisten välillä onkin kaventunut, kun molemmista löytyy ominaisuuksia muun muassa aritmeettisten toimintojen, säädön, raportoinnin ja hälytysten käsittelyn alueilta. Näin ollen ovat myös ohjelmoitavien logiikoiden käyttökohteet lisääntyneet perinteisestä tuotantolinjan tai yksittäisen koneen ohjauksesta aina koko tehdasjärjestelmän hallintaan. Tuotekirjon toisesta päästä kehittyneet kevytrakenteiset ohjauslaitteet ovat puolestaan ottaneet paikkansa esimerkiksi traktorien ja metsäkoneiden toimintojen hallinnassa. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 102)

## 2.1 Logiikkakomponenttien valinta

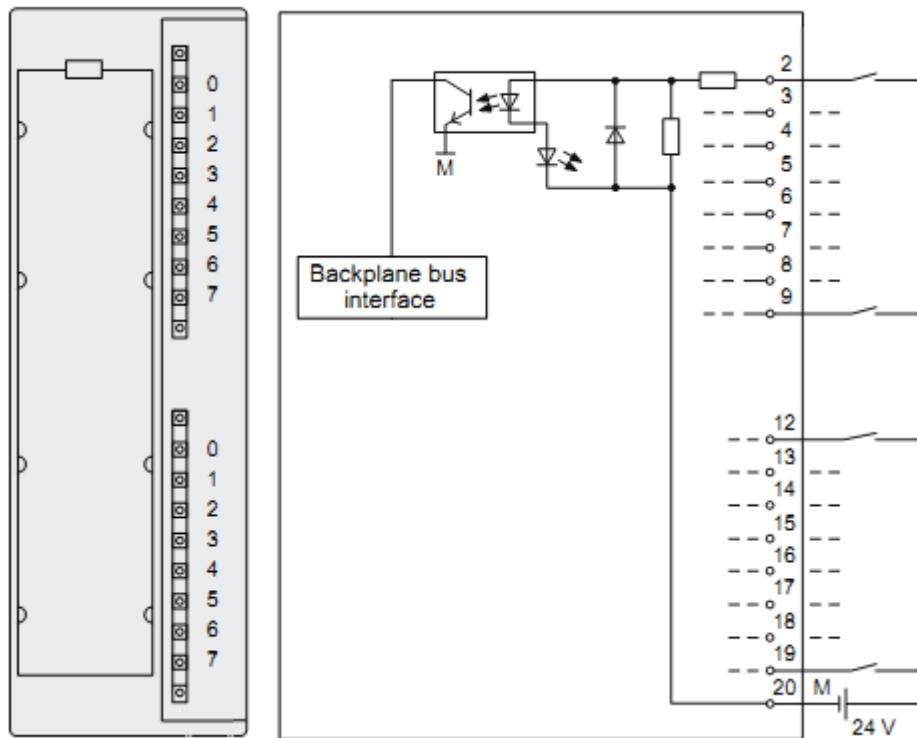
Peruskivinä logiikkakomponenttien valinnassa automaatiotoimitukseen on kenttälaitteiden tarvitsemien analogisten ja binääristen tulojen sekä lähtöjen lukumäärä. Lisäksi on syytä huomioida laitteiston riittävä suorituskky toteutettavien toimintojen monimutkaisuuden tai aikakriittisyyden kanssa, yhteensopivuus jo mahdollisesti

olemassa olevaan järjestelmään sekä uuden laitteiston hinta. Näistä tarvittava I/O-määrä antaa hyvän kehyksen minkä kokoluokan logiikka sopii parhaiten kulloiseenkin tarpeeseen. Loput puolestaan ovat enempi tai vähempi toisensa poissulkevia kriteerejä. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 115-117)

Normaalisti yksittäisen koneen ohjaukseen riittää pieni ja kompakti yhdysrakenteinen logiikka, jotka ovat edullisia mutta monesti rajallisesti laajennettavissa. Suuremmat kokonaisuudet puolestaan instrumentoidaan normaalisti modulaarisilla logiikoilla, joita pystytään räätälöimään pitkälle ohjattavaan järjestelmään sopivaksi. Rakenteesta, valmistajasta tai tuoteperheestä riippumatta kaikista logiikoista löytyy virtalähde, tulo- ja lähtöyksiköt ja näitä ohjaava keskusyksikkö. Viimeksi mainittu sisältää prosessorin ja ohjelmamuistin sekä tavallisesti myös liityntäportin ohjelmointilaitteelle. Edellisten lisäksi laitteistoon voi kuulua valmistajan valikoimista riippuen myös erikoisyksiköitä esimerkiksi nopeaan asemointiin tai väyläliityntöihin. Erikoisyksiköt ovat usein ns. älykkäitä yksiköitä, jolloin ne sisältävät myös oman, ohjattavalle toiminnolle varatun prosessorin. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 115-117)

### 2.1.1 Tuloyksikkö

Tuloyksiköt voidaan jakaa kahteen eri päätyyppiin: binäärisiin ja analogisiin. Näistä ensimmäiseen liitetään kenttälaitteista esimerkiksi kytkimet, eli sen kautta välitetään on/ei-tiedot keskusyksikölle. Binääriset tulot, joita myös diskreeteiksi tuloiksi kutsutaan, ovat nykyään poikkeuksetta toteutettu erottamalla kenttä ja logiikka galvaanisesti toisistaan. Tällä tavoin saavutetaan parempi logiikkalaitteiston suojaus virtapiikeiltä ja häiriöiltä. Erotus toteutetaan yleensä optoerottimin, mutta niiden asemasta on mahdollista käyttää myös esimerkiksi muuntajia tai releitä. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 107-110)

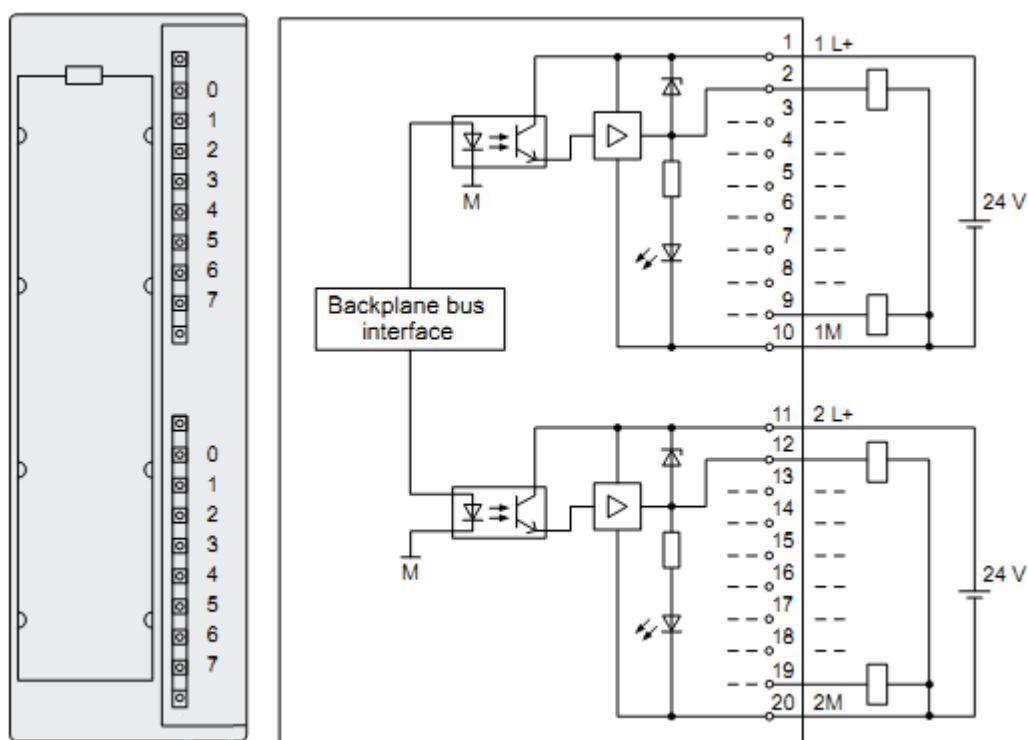


Kuvio 6. Binäärinen tuloyksikkö toimintakaavioineen, SM 321 (Siemens AG 2011)

Analogiset tuloyksiköt on puolestaan kehitetty vastaanottamaan liukuvia anturitietoja, ja näihin tuloihin voidaan kytkeä suoraan esimerkiksi lämpötila- ja paineantureita. Jotta keskusyksikkö pystyisi käsittelemään analogista mittausdataa, on sille tehtävä analogi/digitaali-muunnos tuloyksikössä. Yksiköt on useimmiten mahdollista konfiguroida esimerkiksi DIP-kytkimin käytettävälle standardiviestille, joita ovat 0...20 mA, 4...20 mA, 0...5 V, 0...10 V, -5...5 V ja -10...10 V. Mitattu anturiviesti muunnetaan 8...16 bitin digitaalisiksi ja mitä enemmän bittejä käytetään muunnokseen, sitä parempi resoluutio, eli tarkkuus on mahdollista saavuttaa mitattavalle suurelle. Nopeutta vaativissa sovelluksissa tulee resoluution lisäksi kiinnittää huomiota myös näytetaajuuteen, joka määrittää aika-akselin tarkkuuden. Modernit analogiayksiköt soveltuvat useimpiin käytännön tarpeisiin jo sinällään, joskin aikakriittisyys huomioiden esimerkiksi työstökoneen paikoitus on sellainen tehtävä, jossa voi olla perusteltua käyttää nopeampaa asemoinnin erikoisyksikköä. (Parr 1999, 140-149; Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 107-110)

### 2.1.2 Lähtöyksiköt

Kuten tuloyksiköitä, on myös lähtöyksiköitä sekä binäärisiin että analogisiin tarkoituksiin, ja niihin kytketään kentän kaikki logiikkaohjatut toimilaitteet. Binäärisiin lähtöihin liitetään katkaisinohjattavat laitteet, kuten merkkilamppujen ja magneettiventtiilien virtapiirit sekä moottorien syöttökontaktorien ohjausvirtapiirit. Kuten binääriset tulot, ovat myös lähdöt galvaanisesti erotetut optoerottimin tai relein. Kenttäkytkennän hoitaa joko rele, transistori tai triakki. Edellisistä yleisimmin käytetty on rele, koska sillä voidaan ohjata tasa- ja vaihtojännitepiirejä 250 V:in asti. Toiset kaksi ovat puolijohdekytkimiä, joita puolustaa mekaanista relettä pidempi elinikä. Näiden heikkoutena on kuitenkin rajallinen kuormitusmahdollisuus sekä toiminta ainoastaan omalla jännitealueellaan: transistori 24 VDC ja triakki 230 VAC. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 108-110)



Kuvio 7. Binäärinen lähtöyksikkö toimintakaavioineen, SM 322 (Siemens AG 2011)

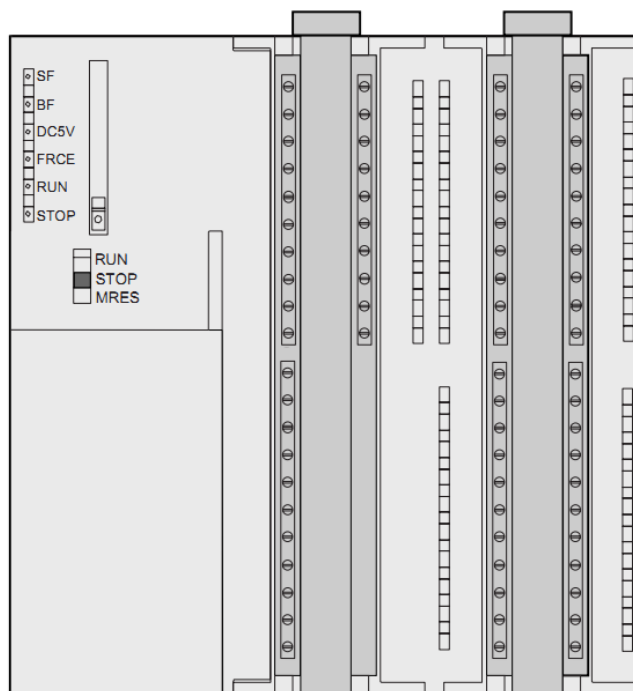
Analogisia lähtöyksiköitä käytetään toimilaitteiden ohjaukseen standardiviestein, ja kohteena voi olla vaikkapa oikosulkumoottorin nopeudenohjaus taajuusmuuttajan avulla. Mikäli edellisessä järjestelmässä olisi käytössä myös nopeustiedon takaisin-



kytkentä, olisi se mahdollisuuksien mukaan laajennettavissa myös säätöpiiriksi. Itse analoginen lähtöyksikkö on rakenteeltaan helppo mieltää analogisten tuloyksiköiden peilikuvaksi. Ehkä hieman reilu yksinkertaistus, mutta käytännössä hyvinkin lähellä totuutta, sillä analoginen lähtöyksikkö suorittaa keskussyksiköltä tulevalle digitaaliselle digitaali/analogi-muunnoksen ja ohjaa sen lähdön standardiviestiksi. Myös nopeuteen ja tarkkuuteen liittyvät seikat pätevät samalla tavoin niin analogiseen lähtö- kuin tuloyksikköön. (Parr 1999, 150-152; Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 108-110)

### 2.1.3 Keskussyksikkö

Ohjelmoitava logiikka on kuin tehtävänsä räätälöity mikrotietokone. Kuten tietokoneissa, myös logiikan keskussyksikössä on sen toimintoja ohjaava prosessori ja käyttöjärjestelmä sekä muistia ja vaihteleva määrä mahdollisia kommunikointiportteja. Prosessoreita voi logiikasta riippuen olla yksi tai useampia ja ne toimivat käyttöjärjestelmän ja ladatun sovellusohjelman puitteissa huolehtien samalla bitti- ja sanaoperaatioista sekä kommunikoinnista keskussyksikön ulkopuolelle. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 107)



Kuvio 8. Keskussyksikkö I/O -kortein, CPU 312 (Siemens AG 2011)

Keskusyksikön muistiin on ladattu myös käyttäjän laatima sovellusohjelma, joka on joukko käskyjä loogisten toimintojen toteuttamiseen tilatietojen mukaisesti. Tilatiedot keskusyksikkö saa lukemalla laitteiston tulot ja reagoi niiden muutokseen asettamalla ohjelman mukaiset lähdöt päälle. Toiminnallisesti keskusyksikkö ja näin ollen logiikka kokonaisuudessaan voi toteuttaa tämän joko pyyhkäisemällä tai reaaliaikaisesti. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 112)

Pyyhkäisevässä logiikassa sovellusohjelma toteutuu sykleissä ja tällöin ohjelmankierrossa on kolme vaihetta. Ensiksi luetaan tulojen tilat tulojen kuvamuistiin. Seuraavaksi suoritetaan ohjelma ensimmäisessä vaiheessa kirjoitetuin tuloehdoin sekä asetetaan lähtöjen kuvamuistiin ohjelmaa vastaavat lähtötilat. Viimeisessä vaiheessa asetellaan logiikan lähdöt lähtöjen kuvamuistia vastaaviin tiloihin. Ohjelmankiertoaika, eli pyyhkäisy on tyypillisesti joitakin millisekunteja, mutta sen aikana tapahtuvat tulojen tilamuutokset eivät vaikuta käynnissä olevaan sykliin.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 112-113)

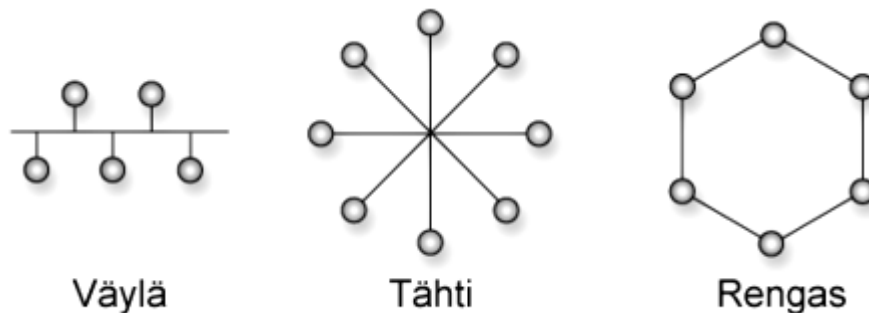
Mikäli logiikan halutaan reagoivan tilanmuutoksiin välittömästi, on käytettävä reaaliaikaista logiikkaa. Reaaliaikaisissa logiikoissa ei käytetä lähtöjen eikä tulojen kuvamuisteja, vaan I/O-tilat voivat muuttua myös ohjelman suorituksen aikana. Ohjelmointi voi kuitenkin olla hankalampaa, sillä toimintojen ajoitus jatkuvasti muuttuvien tilatietojen myötä saattaa aiheuttaa loogisia virheitä järjestelmän toiminnassa. Moderneissa logiikoissa voidaan yleensä valita jompikumpi edellisistä toiminnallisuuksista, tai niitä on joissain tapauksissa myös mahdollista yhdistellä. Näin esimerkiksi työturvallisuuteen vaikuttavat tulot ja lähdöt voidaan käsitellä reaaliaikaisesti ja muu ohjelma pyyhkäisevästi. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 113)

## 2.2 Logiikan hajautus

Logiikan hajautus toteutetaan käyttämällä hajautusyksiköitä. Toiminnan tärkein etu on I/O-korttien tuonti lähelle kenttälaitteita, jolloin ne ovat jaettavissa asemakohtaisesti ja kaapeloinnin määrä sekä kenttäkaapeloinnin pituus pienenevät. Hajautuksella on myös mahdollista estää koko järjestelmän lamaantuminen jos yksi ohjauslaite

vikaantuu sekä helpottaa vian paikantamista ja järjestelmän laajentamista I/O-määrän tai tiedonkäsittelykyvyn lisäämiseksi. Hajautusyksiköt kalustetaan alueella tarvittavin tulo- ja lähtöyksiköin sekä logiikkaväylän liityntäyksiköllä. Lisäksi sillä voi olla myös oma keskusyksikkö prosessoreineen paikallista tiedonkäsittelyä varten. (Parr 1999, 31-33; Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 123-127)

Hajautettua järjestelmää rakennettaessa on valittava hajautusperiaate ja -aste soveluksen tarpeeseen käyväksi. Lisäksi määritetään laitteiden väliset tehtävät, jolloin pystytään välttämään ristiriidat toiminnassa. Hajautusperiaatteena käytetään tyypillisesti isäntä-orjaperiaatetta tai suoria yhteyksiä erilleen sijoitettujen logiikoiden välillä. Edellisten erot ovat laitehierarkiassa ja soveltuvuudessa järjestelmän eri tasoille. Mainituista ensimmäinen valitaan tavallisesti koneen ohjaustasolle, jossa isäntälaitte hallinnoi tiiviisti hajautusyksiköitä jakaen näille tehtäviä. Jälkimmäinen on puolestaan järkevämpi valinta ylemmillä tasoilla, jossa liittyvien laitteiden ohjaamat prosessit eivät enää ole suoraan riippuvaisia toisistaan. Rakenteeltaan hajautuksen topologia voi olla jokin kuvion 9 malleista. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 123-127)



Kuvio 9. Hajautuksen verkkotopologiat

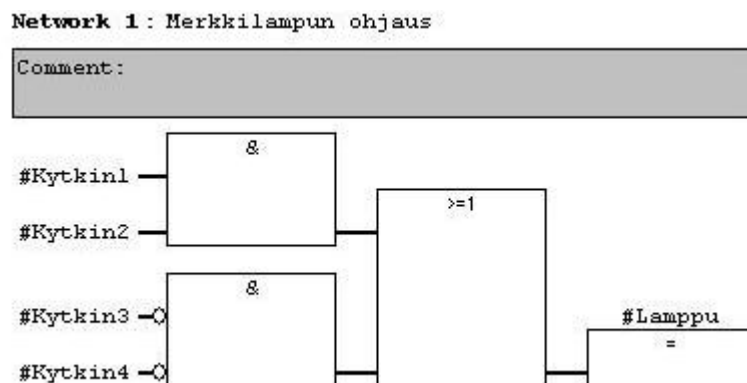
Valitusta hajautusperiaatteesta ja -asteesta riippumatta laitteiden välille tarvitaan myös keino tiedonsiirtoon. Nykyään kuvaan astuu useimmiten automaation kenttäväylä, johon järjestelmän logiikat, hajautusyksiköt, tietokoneet ja enenevässä määrin myös toimilaitteet liittyvät keskitetysti. Eri logiikkavalmistajilla löytyy yleensä tarjota oma versionsa kenttäväylästä, jotka eroavat siinä määrin toisistaan, etteivät ne ole suoraan yhteensopivia keskenään. Tapauskohtaisesti saattaa kuitenkin olla mahdol-

lista käyttää adaptoreita eri valmistajien logiikoiden kesken. Yhteistä kaikille kenttäväylille on se, että siihen liittyvillä laitteilla täytyy olla yksilöidyt osoitteet. Niitä käytetään tiedonsiirron viitteinä osoittamaan minkä laitteiden välillä kommunikointi tapahtuu. Väyläverkkoon liitettävien laitteiden suurin mahdollinen lukumäärä vaihtelee kenttäväylästä toiseen. Tämä ja muut tarkemmat tiedot selviävät parhaiten valmistajan listaamista teknisistä tiedoista. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 123-127)

## 2.3 Logiikan ohjelmointi

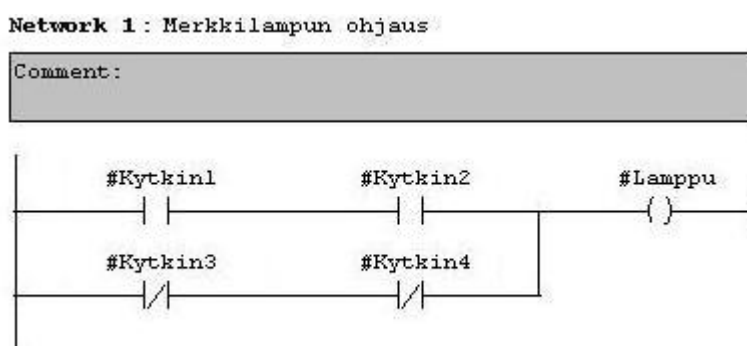
Logiikkaa ohjelmoitaessa suunnittelijan laatimat logiikka- tai relekaaviot kirjoitetaan ohjelmointilaitteella ja siirretään logiikan muistiin. Ohjelma rakennetaan halutun toiminnallisuuden ympärille valitun ohjelmointitavan mukaan, joista yleisimpiä ovat: logiikkakaavio (Function Block Diagram), relekaavio (Ladder Diagram) ja käskylista (Statement List). (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 117)

Logiikkakaavio on rakenteeltaan elektroniikkasuunnittelun kytkentäkaavioita vastaava ja siinä operaatiot esitetään standardoiduin logiikkasymbolein. Sovellettavat standardit ovat IEC 617 (IEC 1082) ja IEC 61131-3, joista ristiriitatapauksessa jälkimmäinen on määräävä. Logiikkakaaviossa operaatiot esitetään toimintoa kuvaavin merkinnöin, jolloin siitä muodostuu tiivis ja havainnollinen. (Parr 1999, 54-56; Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 121)



Kuvio 10. Logiikkakaavio (FBD) esitysmuotona STEP 7-ohjelmassa

Relekaavio on luetelluista ohjelmointitavoista perinteisin. Tämä on perusteltua huomioiden ohjelmoitavien logiikoiden alkuperän juuri releohjauksien korvaajana. Relekaavio-ohjelmointi on määritelty standardissa IEC 61131-3 ja sen esitysmuotoa kutsutaan usein tikapuu- tai kosketinkaavioksi. Ohjelmat rakentuvat JA- sekä TAI- kytkettyistä koskettimista. Ohjelmointitavan käskykanta on ollut alkujaan suppea, mutta uusimmissa ohjelmointityökaluissa kanta on kuitenkin laajentunut pelkistä sulkeutuvista ja avautuvista koskettimista monipuolisemmaksi, sisältäen yleensä samat toiminnot, kuin logiikkakaavio-ohjelmointitapa. (Parr 1999, 51-54; Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 119-121)



Kuvio 11. Relekaavio (LAD) esitysmuotona STEP 7-ohjelmassa

Mikäli on tarvetta toteuttaa monimutkaisia käskyjä, joita ei pystytä ohjelmoimaan muilla esitysmuodoilla, on ne kirjoitettava käskylistäksi. Esitysmuotona käskylistä on vaikeaselkoinen vaikkakin se noudattelee edellisten ohjelmointirakenteita. Ohjelmalohkoista on löydettävissä sen aloittava latauskäsky ja joukko loogisia lukituksia (vrt. relekaavion koskettimet), joiden perusteella ohjataan muistipaikan tai lähdön tilaa. Myös käskylistan tarkempaan määrittelyyn voi tutustua standardissa IEC 61131-3. (Parr 1999, 58-61; Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 122)



Kuvio 12. Käskylistä (STL) esitysmuotona STEP 7-ohjelmassa

Edellisten esitysmuotojen lisäksi standardi IEC 61131-3 määrittelee ohjelmointitavat: strukturoitu teksti korkeamman tason ohjelmointikielien käyttöä varten ja askelkaavion sekvenssiohjauksien luontia varten. Sekvenssiohjaukset ovat siirtoehdoin toteutuvia, askeltavia toimintoja, jossa eri toiminnot voivat tapahtua myös yhtäikaa. Tämä johtaisi perinteisin ohjelmointitavoin haasteellisiin lukitusehtojoukkoihin, jonka vuoksi sekvenssien luontia varten on olemassa omat ohjelmointityökalunsa. Itse sekvenssiohjelman esitystapa noudattaa standardia SFS-EN 60848.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 123; SESKO Ry 2011)

### 2.3.1 Ohjelman rakenne

Logiikkaohjelmat ovat rakenteellisesti toteutettavissa joko lineaarisina tai modulaarisina. Lineaarinen toteutustapa valitaan lähinnä prosessin tai käytettävän logiikan niin erikseen vaatiessa. Sen rakenne on yksinkertainen, sillä lukitusehdoin toteutettavat toiminnot toteutetaan peräkkäin rivi riviltä yhdessä ohjelmalohkossa. Tunnusomaista on ohjelmasyklin kierto kerrasta toiseen tismalleen samanlaisena, vaikka lohkon sisällä onkin mahdollista suorittaa siirtymiä ohjelmakohtien yli haluttuihin toimintoihin. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 152)

Kuten missä tahansa muussakin ohjelmoinnissa, runsas siirtymien käyttö voi tehdä ohjelman seurannasta hankalaa. Mikäli mahdollista, on tällöin parempi valita jo lähtökohtaisesti aliohjelmiin jaettu modulaarinen ohjelmarakenne, jolloin yhden pitkän ketjun sijaa ohjelma koostuu pienemmistä osasista. Modulaarinen muoto mahdollistaa aliohjelmien, eli ohjelmayksiköiden järjestelyn tehtävien mukaisesti ja myös ohjelmakierto saadaan tällä tavoin joustavammaksi. Ylimmällä tasolla olevasta pääohjelmasta tehdään kutsut toimintojen aliohjelmiin, joilla puolestaan voi olla edelleen omia aliohjelmia. Samalla koko ohjelman sijasta osa ohjelmayksiköistä voidaan suorittaa esimerkiksi vain tietyin aikavälein tai ehdoin.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 152)

Mistä ohjelmayksiköt sitten muodostuvat? Ohjelman perusyksiköitä kutsutaan virtapiireiksi, jotka esitetään käytettävän ohjelmointitavan mukaisesti. Kuvioissa 10-12 ilmenevistä ulkonäön eroavaisuuksista huolimatta kaikki ohjelmointitavat toteuttavat kuitenkin täsmälleen saman toiminnan. Virtapiirit rakentuvat loogisista operaatioista (AND, NAND, OR, NOR jne.), joiden lisäksi käytettävissä on myös ajastimia, laskureita, siirtorekistereitä, aritmeettista käsittelyä sekä muita erikoistoimintoja logiikan ominaisuuksien mukaisesti. Ohjelmoinnissa operaatioiden ehtoina käytettävät muuttujat ovat joko ulkoisia (global) tai paikallisia (local). Ulkoiset muuttujat määritellään koko sovellusohjelma-alueelle, jolloin niihin voidaan viitata kaikista ohjelmayksiköistä. Paikalliset muuttujat ovat puolestaan toiminnaltaan ohjelmayksikkökohtaisia. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 119)

### 2.3.2 Ohjelmointilaitteet

Standardin IEC 61131 yhtenä tavoitteena on ollut yhtenäistää eri logiikkavalmistajien ohjelmointilaitteiden ja -ohjelmien yhteensopivuutta, mutta ne ovat pääsääntöisesti yhä logiikka- tai logiikkaperhekohtaisia. Ohjelmointilaitteella on pystyttävä tekemään muutoksia ohjelmalohkojen virtapiirien operaatioihin, siirtämään käyttäjän luomat ohjelmat laitteen muistista logiikan ohjelmamuistiin ja päinvastoin sekä monitoroida ohjelman suoritusta. Laitteet on tavallisesti jaettu kolmeen kategoriaan: kompakteihin käsiohjelmointilaitteisiin, näyttöruudullisiin ohjelmointilaitteisiin sekä PC-tietokoneissa käytettäviin ohjelmointiohjelmiin.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 117-119)

Käsiohjelmointilaitteet näyttävät tavallisesti ohjelmasta vain yhden virtapiirin kerrallaan. Ohjelmointi tapahtuu useimmissa malleissa komento komennolta käskylistamuodossa suoraan logiikan muistiin, ja siten myös ohjelmien tallennusmahdollisuudet ulkoisille medioille ovat huonot. Käsiohjelmointilaitteet ovat kuitenkin edelleen varsin käyttökelpoisia laitteita kunnossapidon seurantavälineiksi vaikkapa logiikan tila- ja diagnostiikkatietojen lukemiseen.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 117-119)

Näyttöruudulliset ohjelmointilaitteet ovat toiminnoiltaan käsiohjelmointilaitteita monipuolisempia, ja ne ovat yleensä valmistajakohtaisia. Niillä voidaan valita ohjelmointitavaksi logiikkakaavio, relekaavio tai käskylistä. Tavallisesti laitteet sisältävät logiikkavalmistajan toimesta myös monipuoliset ohjelmakirjastot monimutkaisempien toimintojen toteuttamista varten. Aikaisimpien näyttöruudullisten ohjelmointilaitteiden pulma oli niiden suurehko koko sekä soveltumattomuus muuhun kuin logiikkaohjelmointiin. Nykyään nämä laitteet pohjautuvat kuitenkin yhä enenevässä määrin PC-tietokoneiden komponentteihin ja käyttöjärjestelmiin, jolloin niitä voidaan käyttää myös aiempaa monipuolisempaan työskentelyyn. Käytännön lisäarvo vaikkapa tavanomaiseen kannettavaan tietokoneeseen verrattuna tehdään räätälöidyllä liitettävyydellä sekä esiasennetuilla ja lisensoituilla ohjelmointityökaluilla.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 117-119)

Kolmesta listatusta viimeisin, eli PC-pohjaisten ohjelmointityökalujen käyttö on nykyään yleisintä. Tähän on varmasti johtanut hiljalleen paitsi tietotekniikan komponenttien hintakehitys sekä kannettavien tietokoneiden helppo liikuteltavuus. Samalla laitteella käy ohjelmoinnin lisäksi myös projektin viestintä-, varmennus- sekä dokumentointitehtävät ja voi samalla tietokoneella olla asennettuna useamman logiikkavalmistajan ohjelmointityökalutkin yhtäaikaisesti. Aivan vapaasti ei tietokoneille voi kuitenkaan kaikkia ohjelmointiohjelmia asentaa, vaan niiden käyttöön tarvitaan yleensä ohjelmakohtainen lisenssiavain. Tämä voi olla joko fyysinen moduuli tai salausalgoritmin tuottama suojauskoodi.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 117-119)

Modernit ohjelmointiohjelmat taipuvat useisiin eri esitysmuotoihin niin ohjelmoinnin kuin dokumentoinninkin osalta. Ohjelmia voidaan kirjoittaa ja muokata joko logiikasta irrallaan (off-line) tai kytkettynä (on-line), jolloin joitakin muutoksia voidaan tehdä jopa pysäyttämättä prosessia. Muita ohjelmointiohjelmista yleisesti löytyviä ominaisuuksia ovat muun muassa: mahdollisuus symbolisten ja kommentoitujen osoitteiden käyttöön, valmiiden ja testattujen ohjelmien tallennuksen ohjelmakirjastoon myöhempää hyödyntämistä varten, monipuoliset analysointityökalut ohjelman toi-



minnallisuudesta sekä työkalut ristiviittauslistojen luomiseen käytetyistä, käyttämättömistä tai kommentoimattomista osoitteista.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 117-119)

## 2.4 Operointipaneelit (HMI, Human Machine Interface)

Siinä missä aiemmin käsitellyt tulo- ja lähtökortit toimivat logiikan rajapintana liitettävän prosessin tiedonkeräykseen ja ohjaukseen, logiikan on muodostettava vastaavanlainen rajapinta myös ihmiseen kokonaisuuden hoitajana. Tällainen rajapinta tunnetaan prosessin käyttöliittymänä ja voidaan toteuttaa useammalla eri tavalla käytettävän laitteiston ominaisuuksien puitteissa.

Tavallisesti toteutus on hoidettu ohjattavan laitteen välittömään läheisyyteen kiinteästi asennetulla ohjauspulpetilla, josta löytyy esimerkiksi toimintaympäristön kytkimet, merkkivalot, peukalopyörät, osoitinnäytöt ja muut tarvittavat instrumentit. Tällaisten pulpettien haittapuolena on nimensä mukainen kiinteys, jolloin toimintojen lisäys tai muuttelu on vaikeaa. Edelleen vaikkapa laitosmuutoksissa suoritettavat siirrot ovat monijohtimisilla runkokaapeleilla logiikkaan kytkeytyvien ohjauspulpettien kohdalla aina suuritöisiä tehtäviä. Ei tule unohtaa että fyysisten instrumenttien lukumäärä asettaa rajoitteita myös ohjauspulpetin koolle. Kokoon vaikuttaa myös se, ettei käyttäjän kannalta selkein ja opastavin sijoittelu toimintaryhmiin ole pinta- alatehokkuudeltaan tavallisesti se kaikkein paras.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 190-191)

Edelliset hankaluudet huomioiden ja tietotekniikan kehitys mukaan lukien on helppo ymmärtää, että erillisillä instrumenteilla kalustettuja ohjauspulpetteja useammin käyttöliittymiksi valitaan nykyään yhä enenevässä määrin näyttöruutupohjaisia operointipaneelleja. Operointipaneelin tyypistä riippuen syöttölaitteena voi olla kosketusnäyttö, yhdysrakenteiset nuoli- ja funktionäppäimet tai Windows-pohjaisissa yksiköissä lähes mikä tahansa pc-tietokoneen ohjauslaite. Paneelien näytöille on mahdollista tuoda säädösten vaatimia kovia lukituksia lukuun ottamatta tismalleen samat hallintalaitteet virtuaalisina, jotka on perinteisesti täytynyt upottaa ohjauspulpettiin.

Myös muutosten tekeminen näyttöihin on helpompaa ja käytännön lisäarvoa kehittyneempien operointipaneelien sovelluksissa tuo esimerkiksi ohjattavan prosessin runsaat visualisointimahdollisuudet. Havainnollisempaa onkin vaikkapa näköyhteyden kantamattomissa olevan kuljettimen pyöriminen eteenpäin näyttöruudulla, kuin yksinkertainen ”kuljetin käy”-merkkivalo.

(Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 190-191)

Operointipaneeleita on tarjolla logiikkavalmistajakohtaisesti ja mallistosta löytyy tuotteet likimain kaikkiin kuviteltaviin teollisuusympäristöihin. Paneelia valittaessa on huomioitava näyttöruudun riittävä koko käytön havainnollisuuden kannalta, kotelointiluokka suojaamaan asennusympäristön riskeiltä, tarvittava suorituskky laajenusvaroin sekä liitettävyyys ja yhteensopivuus muuhun logiikkalaitteistoon. Kommunikointi logiikan ja operointipaneelin välillä tapahtuu yleisimmin sarjaliikenneväylässä, joskin automaatiolaitteiden yleistä kehitystä seuraten myös operointipaneeleista löytyy jo usein ethernet-liityntämahdollisuus. Näistä menetelmistä sarjamuotoinen liikennöinti onnistuu poikkeuksetta eri laitevalmistajienkin logiikoiden ja operointipaneelien kesken, mutta ethernet-kommunikointi tapahtuu yleensä valmistajan oman protokollan puitteissa rajoittaen näin yhteensopivuutta.

### 3 OHJELMOINTIYMPÄRISTÖT

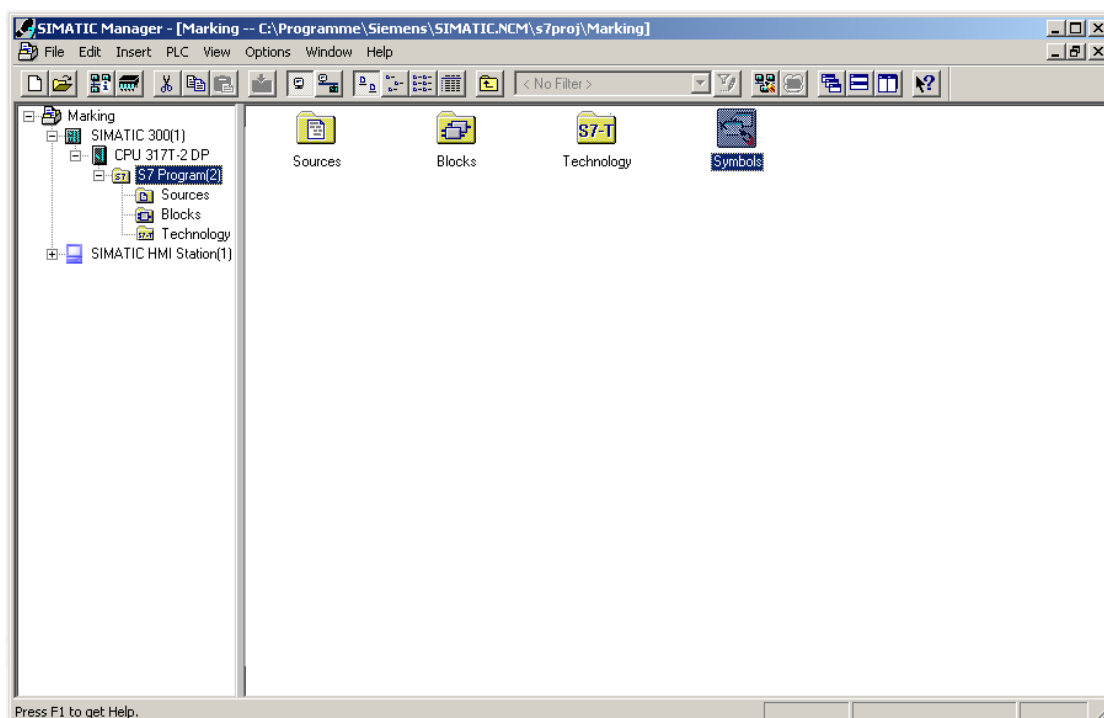
Saatuamme yleiskäsityksen tehtävän luonteesta sekä ohjelmoitavista logiikoista toteutuslueustana, on vuorossa työssä käytettävien ohjelmointisovellusten tarkastelu. Koska työ toteutetaan Siemens SIMATIC 7 -sarjan logiikkalaitteilla sekä saman valmistajan operointipaneeleilla, on ohjelmistot valittava laitteistojen kanssa yhteensopiviksi. Näin ollen logiikkaohjelmointiin käytetään Simatic Step 7 Professional suunnitteluohejelmistoa ja operointipaneelien ohjelmoinnissa työkaluna käytetään WinCC flexible 2008 ohjelmistoa.

### 3.1 SIMATIC Step 7 Professional

Kaikki Siemensin SIMATIC 7 -sukupolven logiikkalaitteet lukuun ottamatta valmistajan suorituskyvyltään keveimpiä LOGO!- sekä sarjan S7-200-ohjaimia konfiguroidaan ja ohjelmoidaan SIMATIC Step 7 -sovellusympäristössä. Perusversio Step 7 -ympäristöstä tukee ohjelmointia kolmella IEC-61131-3 standardin mukaisella kielellä: relekaaviolla (LAD), logiikkakaaviolla (FBD) ja käskylistalla (STL). Perusversio itsessään soveltuu jo kokoluokaltaan niin pieniin kuin suuriinkin automaatioprojekteihin, jonka lisäksi erikoisempia sovelluksia varten ohjelmistoa voidaan vielä laajentaa erilaisilla lisäpaketeilla.

(Siemens AG 2011)

Step 7 Professional on ohjelmiston laajempi versio, johonka on integroitu valmiiksi muutamia Step 7 perusversion lisäpaketteja. Ohjelma sisältää simulointiominaisuu-  
det, eli sovelluksen toimintaa voidaan testata jo ohjelmointikoneessa (S7-PLCSIM).  
Lisäksi Step 7 Professional pitää sisällään kaksi ohjelmointikieltä enemmän: Sekvens-  
siohjelmointiin tarkoitettu (S7-GGRAPH) ja monimutkaisten algoritmien ohjelmointia  
helpottava strukturoitu teksti (S7-SCL). (Siemens AG 2011)



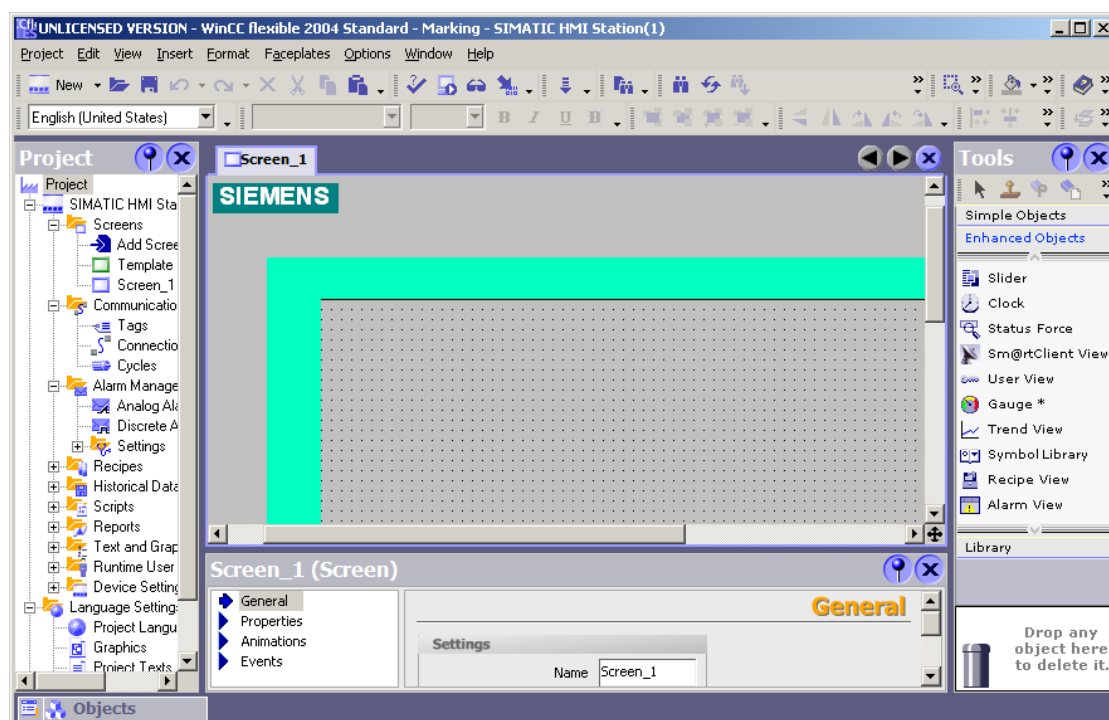
Kuvio 13. SIMATIC Manager

Step 7 ohjelmointiympäristön sydän on SIMATIC Manager. Sen avulla tehdään kaikki projektin määrytykset, ohjelmointi sekä hallinta. Ympäristöstä löytyy työkalut logiikkakomponenttien ja väylärakenteen määrytykseen, symbolilistojen käsittelyyn, sovel-lusohjelmointiin ja testaukseen sekä projektien varmentamiseen ja taltiointiin. Lisäksi SIMATIC Manager integroi kaikki käytettävät järjestelmät, kuten logiikkaohjaimet ja operointipaneelit saman projektirakenteen alle, jolloin myös kaikki niihin liittyvät sovellukset ja ohjelmat aukeavat SIMATIC Managerin kautta. Kuviossa 13 on auki ympäristön pääikkuna, joka ryhmittelee Windowsin resurssienhallinnan tapaan va-sempaan ikkunaan kansiomaisesti projektin rakenteen ja vastaavasti valitun kom-ponentin sisältö näytetään eritellysti viereisessä ikkunassa.

### 3.2 WinCC flexible 2008

Siinä missä edellä esitelty ohjelmointiympäristö keskittyi itse logiikkalaitteisiin, on WinCC flexible Siemensin mukaan "tuote koneläheisen visualisoinnin ohjelmointiin." Se on siis sovellus HMI -rajapinnan luontiin ja työn näkökulmasta WinCC flexiblellä tullaan luomaan valvomo-ohjelma SIMATIC-operointipaneeliin. Yleisesti se sopii käy-tettäväksi pienien Micro-paneelien ohjelmoinnista aina PC-valvomosovelluksien ke-hittämiseen saakka. (Siemens AG 2011)

Monikäyttöisyyttä laajan laitekirjon ohella lisäävät myös mahdollisuus valita käytet-täväksi jokin monikielisen ohjelmointiympäristön 32:sta eri kielestä ja se, ettei ohjel-ma ole millään tavoin teollisuusalasidonnainen. Ohjelmoinnin kannalta WinCC flexi-ble tarjoaa apuvälineiksi esimerkiksi mahdollisuuden valmiiden kirjastojen ja näyttö-elementtien käyttöön. Edelleen tuetaan myös eri tekstien tuonti/vienti-toimintoja ja useita visualisointitapoja. Näistä jälkimmäisiä edustavat myös Sm@rtAccess sekä Sm@rtService, joidenka toiminnot valmistaja nivoo yhteen SM@-konseptiksi. Tämä konsepti avaa muun muassa mahdollisuuden käyttää TCP/IP-protokollaa muuttujien ja näyttöjen siirtotienä. Lisäksi WinCC flexiblen integraatio muihin SIMATIC-tuotteisiin antaa hyvät lähtökohdat koko järjestelmän keskinäiseen tiedonvälityk-seen. (Siemens AG 2011)



Kuvio 14. WinCC flexible

WinCC flexible 2008 on siis ohjelmointiympäristö, jolla luodaan Siemensin operointipaneeleissa suoritettavia sovelluksia. Kuviossa 14 on WinCC flexiblen editorin pääikkuna. Ulkoasultaan ohjelmiston 2008 versio ei eroa kuvion 2004 versiosta, vaan eroavaisuudet ovat toimintojen laajuudessa. Editorin vasemmasta laidasta löytyvät projektin kehittämiseen liittyvät toiminnot, jotka aukeavat välilehdiksi keskelle editoria. Alhaalta keskeltä löytyy graafisten elementtien toiminnallisuuteen vaikuttavat määrittelyt ja oikeaan reunaan on sijoitettu paneeli- ja projektikohtaisesti käytettävissä olevat työkalut.

## 4 OHJELMOINTI

### 4.1 Vanhan ohjelman konversio

Projektin sovellusohjelmaa ryhdyttiin rakentamaan vanhan logiikkaohjelman konversiona eli käännöstyönä. Koska tarkoitukseen sopivaa työkalua, jolla konvertointi olisi onnistunut automaattisesti ei ole olemassa, ohjelmoitiin koko käännöstyö alusta asti uudelleen käsin. Vanhassa järjestelmässä Allen Bradley PLC2-logiikoita oli viisi ja

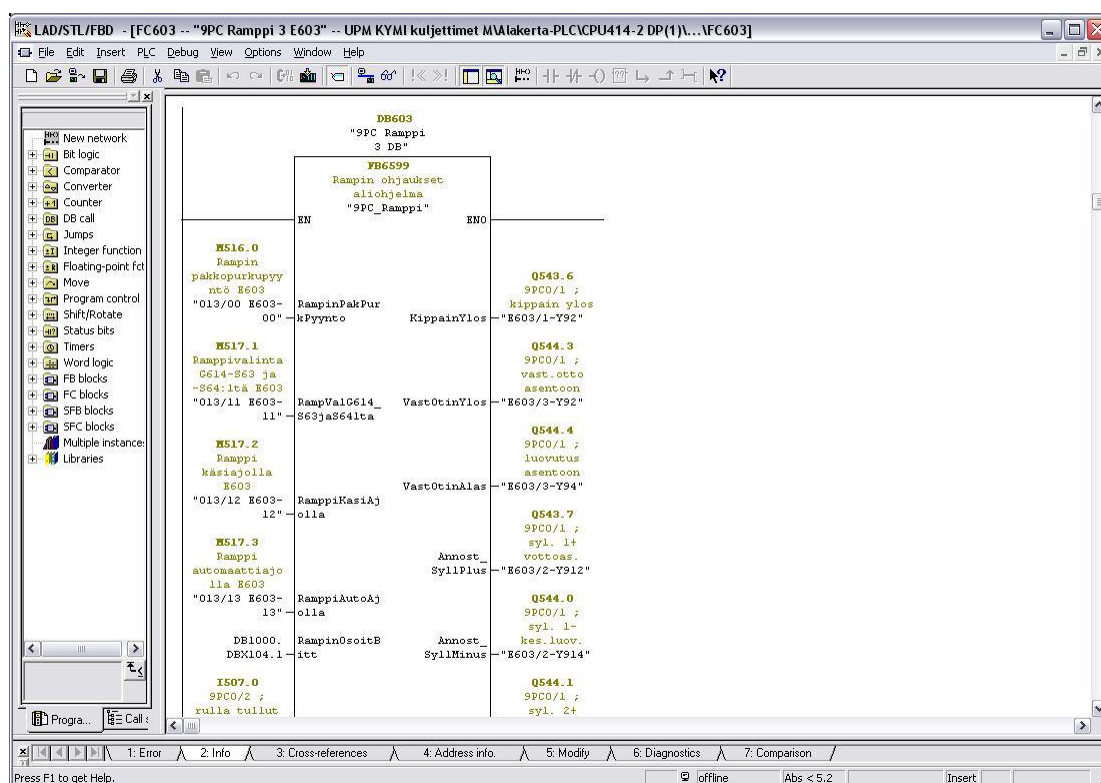
näistä jokaisesta oma relekaaviomuotoinen ohjelmalistauksensa. Työn rajauksen myötä rampiston ja rullanpakkaus koneen jälkeisten kuljettimien alueille, jossa myös rulladatan siirto tapahtuu, jäi kaksi logiikkaa. Nämä tosin olivatkin sitten ohjelmakooltaan suurimmat: kuljetinlogiikan ohjelmassa 1368 virtapiiriä ja rampiston ohjelmassa 2200 virtapiiriä.

Tässä vaiheessa on hyvä huomata vanhan ja uuden järjestelmän eroavaisuudet ohjelmarakenteidensa osalta - PLC2:ssa koko ohjelma on kirjoitettu yhteen pitkään listaukseen, kun taas perinteistä Simatic 7 ohjelmaa edustava uusi sovellusohjelma tulee jakautumaan toiminnallisiin lohkoihin. Lohkojen sijaan PLC2:den ohjelmassa joustavuutta ohjelmakiertoon on aikanaan tehty muun muassa käyttämällä siirtymiä ohjelman sisällä: JMP (Jump to label, hyppy merkkilippuun), JSR (Jump to SubRoutine, hyppy aliohjelmaan), RET (Return, paluu), sekä valinnaisin ehdoin suoritettavin ryhmäohjauksin: ZCL (Zone Control Last state, ryhmäohjaus - ohitus/valinta) ja MCR (Master Control Reset, ryhmäohjaus - pysäytys).

Ennen ohjelmointityön aloittamista jaettiin I/O-, muisti-, laskuri- ja ajastinalueet korvattavaa logiikkalaitteistoa mukaillen. Ohjelmointi tehdään relekaaviona (LAD), koska se on myös alkuperäisen ohjelman esitysmuoto. Sovittiin lisäksi ohjelmalohkojaon pohjautuvan alueiden fyysisiin laitteisiin, jolloin periaatteessa uudessa logiikkaohjelmassa jokaiselle laitteelle tulisi oma ohjelmalohkonsa. Vanhentunut versio toimitusta ohjelmalistauksesta sisälsi niukahkon ohjelmakomentoinnin, jonka avulla ohjelmalistauksesta oli kuitenkin kohtuudella tunnistettavissa mikä osa koodista kuului minkäkin laitteen ohjaukseen. Niinpä vanhaa koodia mukaillen ohjelmaan muotoutui vähitellen tavanomaista hierarkkista ohjelmaa muistuttava rakenne: syklistä pääohjelmasta kutsutaan ohjausalueita, jotka edelleen tilansa mukaan kutsuvat laitteiden sekä toimintojen lohkoja ja niin edelleen.

Pääsääntöisesti konvertointityö eteni virtapiiristä virtapiiriin ohjelmoiden. Vanhassa ohjelmalistauksessa esiintyi paikoitellen kuitenkin niin huomattavia määriä toistoa sekä ohjelmakierron kannalta tyhjää apumuuttujien käyttöä, että näiden ohjelmankohtien osalta katsottiin ajallisesti sekä uuden ohjelman selkeyden kannalta parhaak-

si toteuttaa ne omina toiminnallisina lohkoinaan. Hyvänä esimerkkinä edellisestä käy rullarampiston yksittäisten ramppien ohjaukset. Kun rampin toimintaa ohjaava koodi ohjelmoitiin kertaalleen Step7:n toimintayksiköksi (FB, Function Block), sitä on mahdollista käyttää eri ramppien ohjelmakutsuissa parametroituna kyseessä olevan rampin tulo- ja lähtötiedoin.



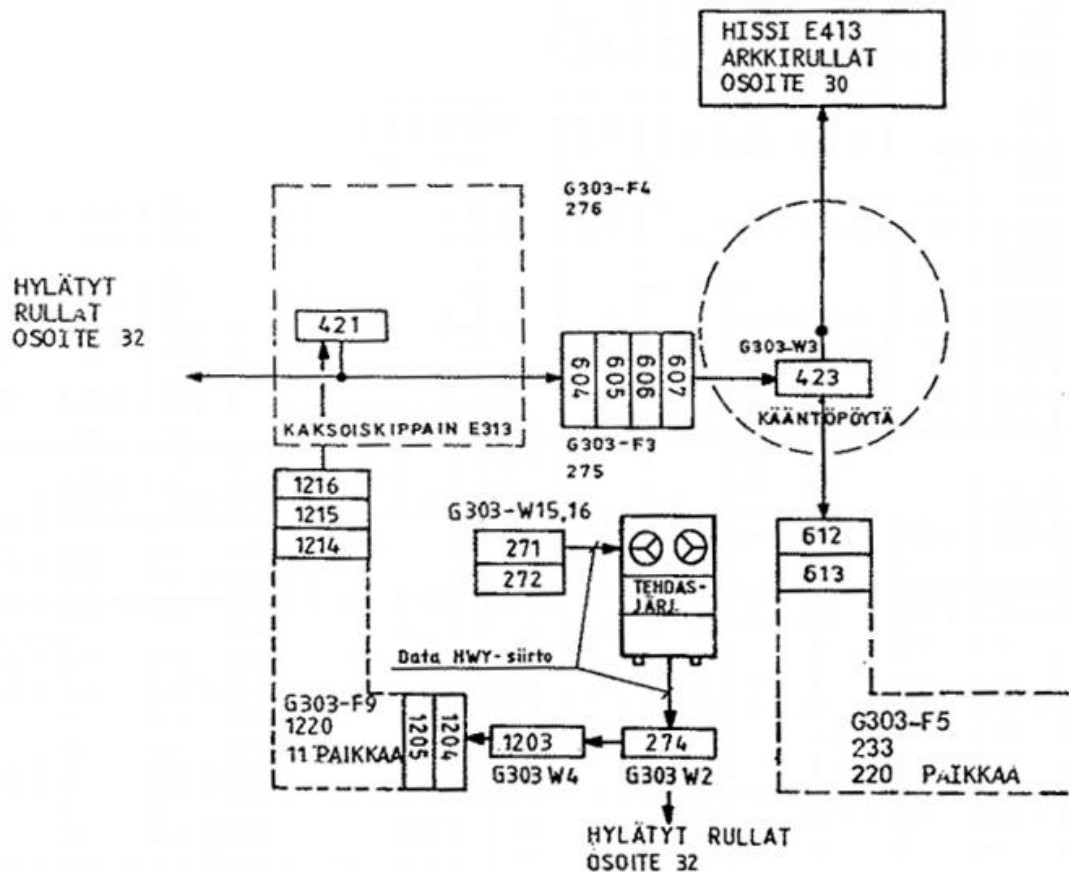
Kuvio 15. Ramppiohjauksien ohjelmakutsu

Kuviossa 15 esitetään osa ramppien toimintayksikön ohjelmakutsusta, jossa parametrit syötetään kehyksen jalkoihin - tulot vasemmalle puolelle ja lähdöt oikealle. Parametrit voivat olla tyyppiä IN, OUT tai IN\_OUT, ja ne on määriteltävä ohjelmaeditorissa, jotta niitä voidaan käyttää. Nyt parametrit toimivat siis toimintayksikön paikallisina muuttujina sekä liitoskohtana ohjelmakutsuun ja siten kyseessä olevan laitteen ohjaukseen. Kuviossa 16 on esillä osa määritellyistä parametreista ja toimintaa ohjaavasta koodista.

## 4.2 Rulladatan siirto

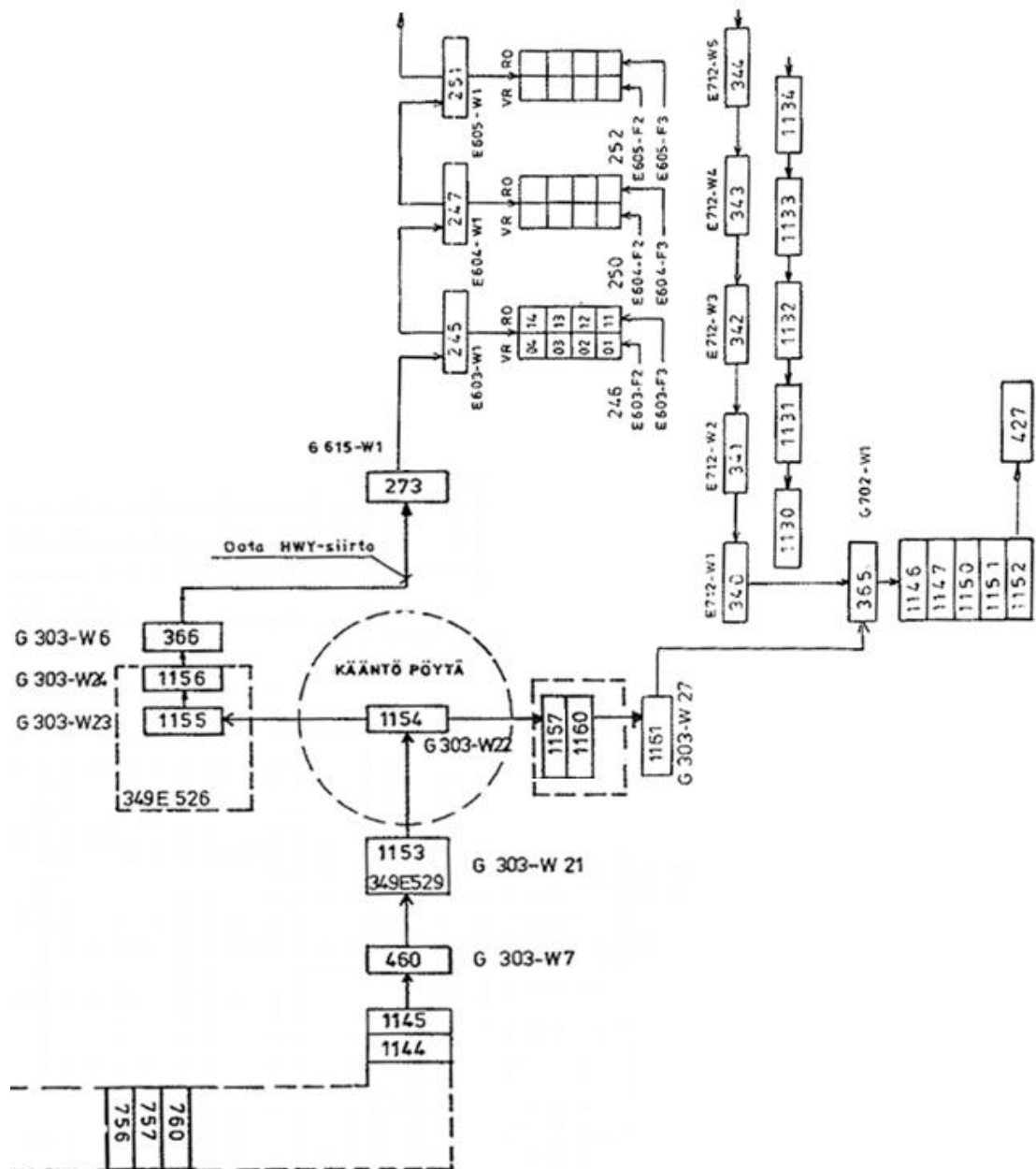
Rullanpakkaus koneelta poistuva rulla viimeistellään etiketillä, josta ilmenee paperi-  
laadun ja fyysisten suureiden lisäksi muun muassa tehdas-, asiakas- ja tilauserätieto-  
ja. Ihmissilmällä helposti tulkittavat etikettitiedot täytyy saada laitteita ohjaavalle  
järjestelmälle sähköisessä muodossa, mikäli varasto-operaatioita halutaan tehostaa  
automaatiota hyväksikäyttäen. Jos kustannuskysymyksiä ei huomioida, voitaisiin eti-  
kettien rulladata lukea jokaiselle yksittäiselle rullankäsittelylaitteelle tullessa esi-  
merkiksi kameroiden tai viivakoodinlukijoiden avulla. Tässä työssä rulladataa kuiten-  
kin siirretään muistipaikasta toiseen eräänlaisena sähköisenä varjona. Data siirtyy  
logiikan sovellusohjelmassa eteenpäin sitä mukaa, kun rulla etenee fyysisesti käsitte-  
lylaitteelta toiselle.





Kuvio 17. Rulladatansiirto kuljettimilla, rullanpakkauskoneen alue

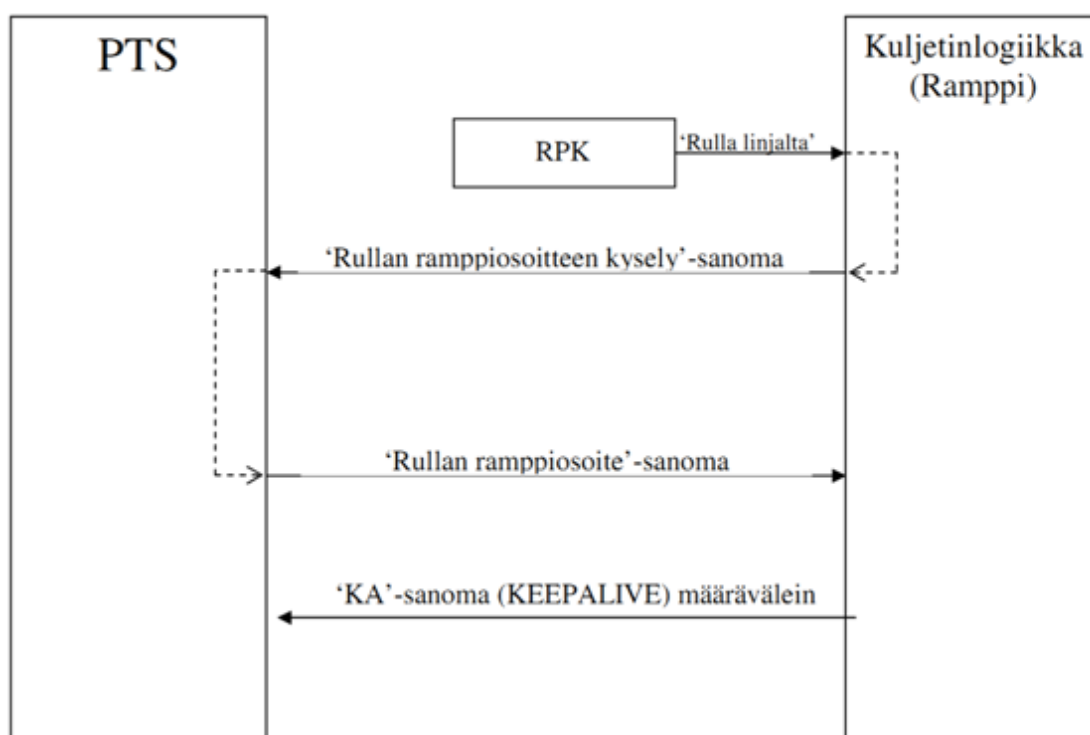
Muistipaikat on sidottu kuljetinjärjestelmän layoutiin, ja ohjelmallisesti niiden toiminta perustuu käsittelylaitteiden rajakytkintietoihin. Kokonaisuuden hahmottamiseksi koko rullankäsittelylaitteiston yhtenäinen layout on esillä liitteessä 1. Kuvioissa 17-18 esiintyvissä positiotunnuksissa ilmenevät kirjaimet tarkoittavat datasanoja "W" ja FIFO (First In, First Out - ensimmäisenä sisään, ensimmäisenä ulos) -rekistereitä "F". Yksittäistä datasanaa käytetään esimerkiksi kääntöpöydällä, jossa ei voi olla kuin yksi rulla kerrallaan. Kuljettimille puolestaan mahtuu useampiakin rullia, joten logiikkaohjelmassa niiden toimintaa mallinnetaan järjestyksessä purkautuvina FIFO -rekistereinä. Modernisoinnin myötä kaikki datasanat ja FIFO -rekisterit on määritelty uuden ohjelman tiedostoyksiköihin (DB, Data Block) seurannan ja muokkauksen helpottamiseksi.



Kuvio 18. Rulladatansiirto kuljettimilla, rampiston alue

Mitä rulladatan tehtäviin sitten kuuluu? Tämä data pitää sisällään muun muassa rullan ramppiosoitetiedon, eli se on avainasemassa rampiston oikean toiminnallisuuden kannalta. Tilauseriin lajittelun ja pinoihin koostamisen lisäksi rulladata ohjaa myös rullankäsittelylaitteita jo ennen rampistoa. Kuviossa 17 esitetään esimerkiksi hylky- ja arkkirullien poikkeava käsittely kuvion 18 rampistolle kuljetettaviin rulliin verrattuna. Tilansäästön vuoksi kuvioista 18 on rajattu näkyväksi kääntöpöydältä alkavan leveiden rullien erikoisrampin lisäksi vain rampit 3-5. Rampia 2 ei järjestelmässä ole.

Rulladataa siirrettiin ja käsiteltiin tietenkin myös korvautuvassa PLC2-järjestelmässä, joten osa toiminnallisuudesta syntyi vanhan ohjelman konversiona. Vaikka runko olikin vanhasta ohjelmasta, sovelluskoodia jouduttiin jalostamaan järjestelmien eroavaisuuksien takia. Rulladatan siirto alkaa rullanpakkaus koneelta ja päättyy rulla-rampistolle. Koska kyseessä on projektin kannalta merkittävä osa ohjelmointia, tul- laan edellä esittelemään pelkästään sen pääpiirteet menemättä tarkempiin yksityis- kohtiin.



Kuvio 19. Sanomanvälitys, PTS-kuljetinlogiikka

Tässä työssä tehdasjärjestelmätasolla toimiva ABB PTS (Product Tracking System, tuotteiden seurantajärjestelmä) -tietokone laatii pinorekisterit ja jakaa niiden mukaiset ramppiosoitteet yksittäisille rullille. Kuvion 19 kaaviota seuraamalla huomaamme, että kun kääritty rulla on poistumassa rullanpakkauslinjalta (RPK), logiikka pyytää rullan ramppiosoitteen lähettämällä kyselysanoman PTS-tietokoneelle. Tämä kyselysanoma pitää sisällään myös käyttäjän ohjauspulpetista tekemiä valintoja, kuten yksittäisten ramppien ja arkkirullahissin käytössä/ei käytössä -tilat sekä rullan hylkäysvalinnan. Kyselysanoman tiedostoyksikkö on esitelty kuviossa 20.

Address	Name	Type	Comment
0.0		STRUCT	
+0.0	RKA_PTS_Sanoma	STRUCT	Rullan ramppiosoitteen kysely
+0.0	Sanomatunnus	ARRAY[1..2]	Sanomatunnus 'KY'
*1.0		CHAR	
+2.0	Data1	STRUCT	Ramppien 1-8 tilat
+0.0	Ramppitila	ARRAY[1..8]	'1'=ramppi käytössä, '0'=ramppi ei käytössä
*1.0		CHAR	
=8.0		END_STRUCT	
+10.0	Data2	STRUCT	Ramppien 9-16 tilat
+0.0	Ramppitila	ARRAY[9..16]	'1'=ramppi käytössä, '0'=ramppi ei käytössä
*1.0		CHAR	
=8.0		END_STRUCT	
+18.0	Data3	STRUCT	Ramppien 17-19 tilat, 20-24 varalla
+0.0	Ramppitila	ARRAY[17..24]	'1'=ramppi käytössä, '0'=ramppi ei käytössä
*1.0		CHAR	
=8.0		END_STRUCT	
+26.0	Data4	STRUCT	
+0.0	Varalla0_4	ARRAY[0..3]	
*1.0		CHAR	
+4.0	Arkkihiissin_tila	CHAR	'1'=käytössä, '0'=ei käytössä
+5.0	Rulla_hylkeys	CHAR	'1'=hylätty linjalla -> lattialle, '0'=ei hylätty
+6.0	Rosoitteen_uudelleenky	CHAR	'1'=on, '0'=ei
=8.0		END_STRUCT	
=34.0		END_STRUCT	
+34.0	Varalla	ARRAY[1..66]	
*1.0		CHAR	
=100.0		END_STRUCT	

Kuvio 20. Rullan ramppiosoitteen kyselysanoma

PTS vastaa kyselyyn lähettämällä "Rullan ramppiosoitte"-sanoman logiikalle. Sanoman tiedostoyksikkö on esitelty kuviossa 21. Seuraavaksi PTS-tietokoneelta saatu ramppiosoitte-sanoma käännetään logiikkaohjelmassa kuvion 17 datasanaan "G303 W2", joka vastaa fyysiseltä sijainniltaan rullanpakkausaseman jälkeistä välipysäytintä.

Address	Name	Type	Comment
0.0		STRUCT	
+0.0	PTS_RKA_Sanoma	STRUCT	Rullan ramppiosoitte
+0.0	Sanomatunnus	ARRAY[1..2]	Sanomatunnus 'RO'
*1.0		CHAR	
+2.0	Ramppinnumero	ARRAY[1..2]	1-19 ('01'=ramppi 1, '30'=arkkirullahissi)
*1.0		CHAR	
+4.0	Rullan_hylkeys	CHAR	'1'=kyllä, '0'=ei
+5.0	Pakkopurku	CHAR	'1'=kyllä, '0'=ei
+6.0	Eran_ensim_rulla	CHAR	'1'=kyllä, '0'=ei
+7.0	Etik_suunta_arkkirull	CHAR	'1'/'0'
+8.0	Eran_viim_rulla	CHAR	'1'=kyllä, '0'=ei
+9.0	Osoite_maaritelty	CHAR	Aina '1'
=10.0		END_STRUCT	
+10.0	Varalla	ARRAY[1..90]	
*1.0		CHAR	
=100.0		END_STRUCT	

Kuvio 21. Rullan ramppiosoitte-sanoma

Rullanpakkausaseman jälkeinen välipysäytin on siis paikka, jossa käsittelylaitteelta toiselle etenevä rulla saa aiemmin mainitun sähköisen varjonsa, eli yksilöllisen rulladatan. Sisällöltään rulladata muodostuu seuraavanlaiseksi datasanaksi:

Bitit 0-4	Rullan osoite binäärisenä: esim. 00001=Ramppi 1, 11110=Arkkirulla jne.
Bitti 5	Hylkyrulla
Bitti 6	Pakkopurkurulla: PP-bitti
Bitti 7	Erän 1. rulla: UR-bitti
Bitti 8	Arkkirullan pyöritysuunta, 1=Uusi varasto
Bitti 9	Erän viimeinen rulla: VR-bitti
Bitti 10	Koko osoite saatu

Kommunikointi tapahtuu modernisoinnin myötä sarjaliikenteen sijasta TCP/IP -protokollaa sekä logiikan päässä Step 7:n AG\_SEND ja AG\_RECV lohkoja käyttäen. Logiikkaohjelman ohjelmalohkoissa rulladataa käsitellään, siirretään ja nollataan Step 7:n sanankäsittelytyökalujen avulla.

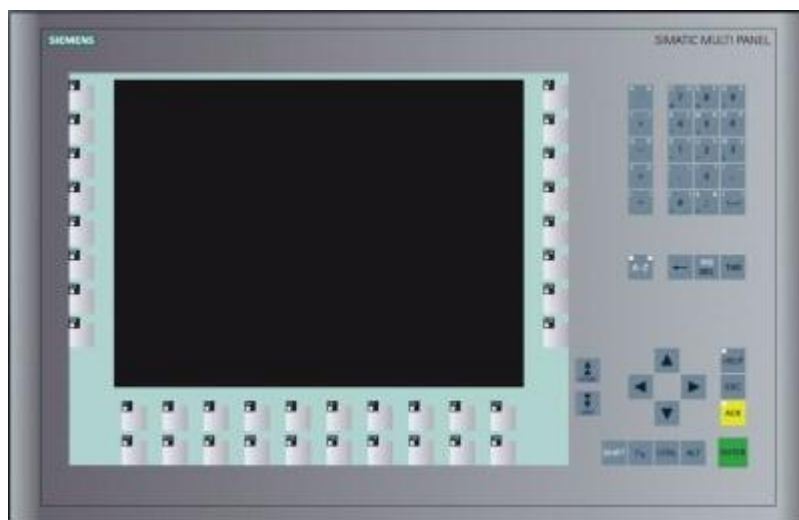
### 4.3 Valvomo-ohjelmointi

Yhtenä modernisointiprojektin osana on lisätä logiikkalaitteiston yhteyteen operointipaneeli käyttöhenkilökuntaa varten. Väylätekniikan ansiosta se voidaan asentaa logiikasta erilleen ja tulee tässä tapauksessa sijoittumaan rullarampiston valvomoon. Asiakkaan toiveesta operointipaneelin tarkoituksena ei ole korvata prosessiohjausta, vaan kuljettimia ja muita laitteita tullaan edelleen ohjaamaan pulpettien vanhoihin kytkimiin. Ohjauksen sijaan paneelin on tarkoitus näyttää järjestelmästä runsaammin tilatietoja sekä parantaa tietojen havainnollisuutta ja muokattavuutta. Tämä toteutetaan visualisoimalla kuljettimia, rullia ja laitetoja sekä opastavalla käyttöliittymällä.

Koska operointipaneelia tai -paneeleita ei ollut olemassa ennestään, työhön ryhdyttiin suunnittelemalla yhdessä asiakkaan kanssa tulevan toiminnallisuuden määrä ja valitsemalla käyttöön sopiva laitteisto. Suunnittelun perusteella operointipaneelin

näytöille ohjelmoidaan järjestelmän tilatiedot kerroksittain layout-kuviin, rampiston rajakytkintiedot ramppikohtaisesti, rulladata rampeilla sekä pino- ja FIFO -rekisteritiedot. Tämän lisäksi operointipaneeliin kirjataan myös järjestelmän diagnostiikka- ja hälytystietoja.

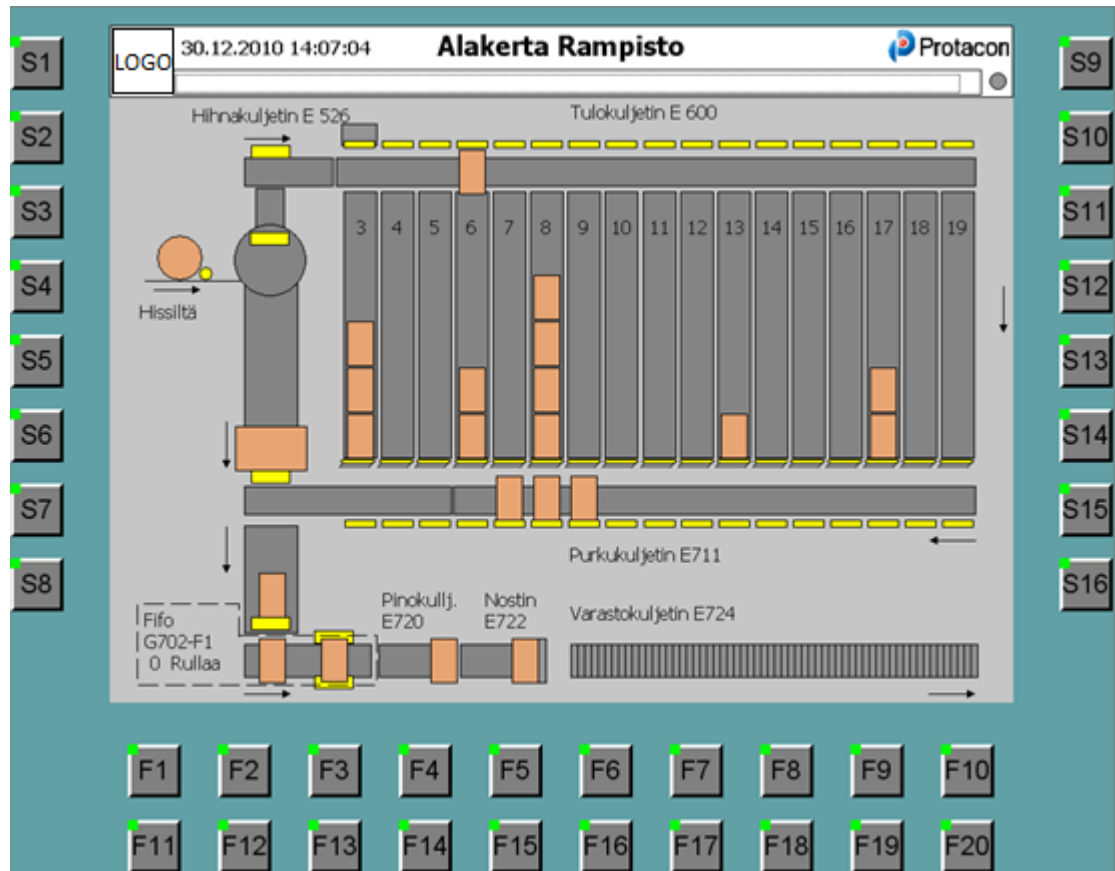
Laitteiston osalta operointipaneeliksi valikoitui painonapein ohjattava Siemensin MP 377 12" Keypad -paneeli. Kosketusnäytöllisten laitteiden hylkäykseen johti mahdollisen riskin välttäminen käyttökeltomaksi likaantumisesta tai naarmuuntumisesta. Valittu paneeli IP 65 -luokiteltuine koteloineen puolestaan kestää jo hieman haasteellisemmankin asennusympäristön. Ohjelmistopuolelta paneelin yhteydessä käytetään WinCC Flexible 2008 SP1 + HF3 -ympäristöä. (Siemens AG 2011)



Kuvio 22. Operointipaneeli MP 377 12" key (Siemens AG 2011)

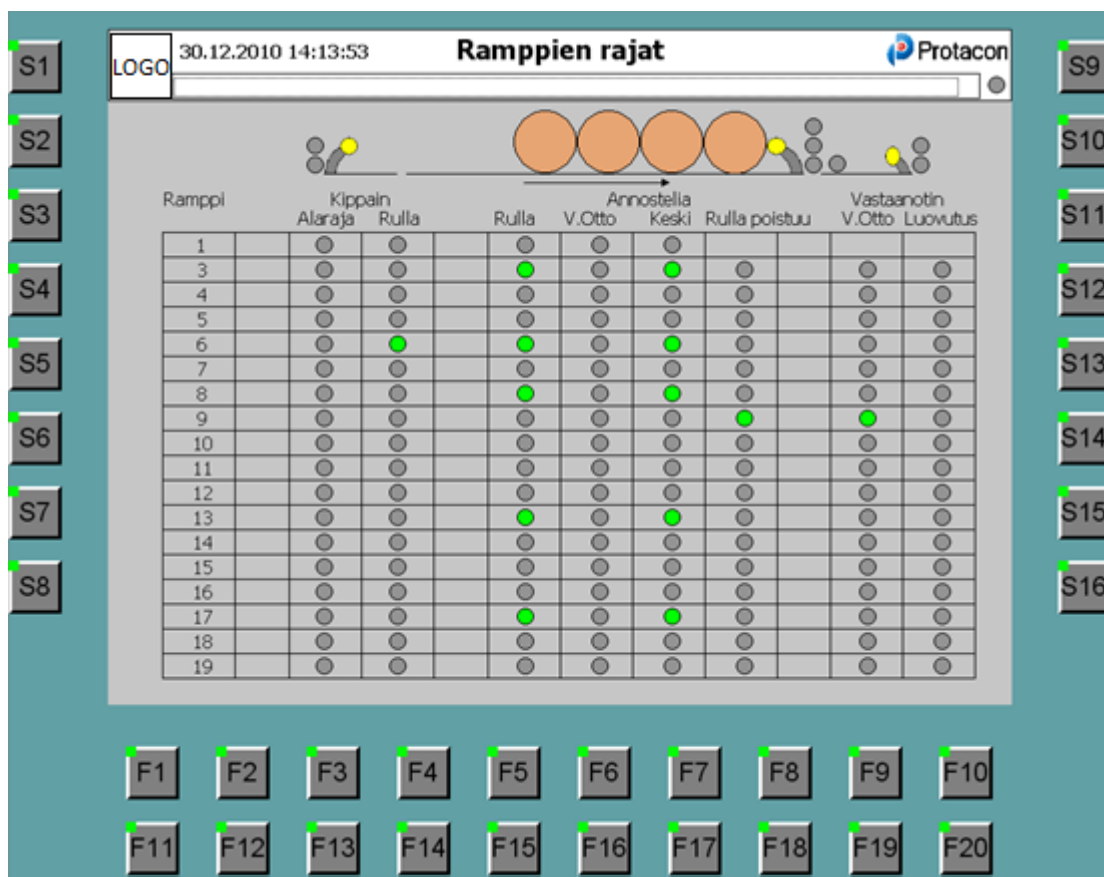
Paneeliohjelmointi alkaa näyttöjen suunnittelulla, jolloin luodaan ulkoasu halutuin symbolein ja ikonein. Seuraavassa vaiheessa tehdään edellisten toimintaan ja visualisointiin vaikuttavat tagit, eli liityntäpisteet, jotka toimivat linkkeinä ohjelmoitavan logiikan ja symbolien välillä. Nyt kun tagit on luotu ja yhdistetty haluttuihin näyttöelementteihin, huomataan operointipaneelin näytöllä logiikan tilatiedon vaihtuminen vaikkapa värin, kuvan tai näkyvyyden/näkymättömyyden muutoksena. Seuraavaksi esitellään asiakkaan toiveiden mukaisiksi hiotut näytöt sekä niiden toiminnallisuudet.

#### 4.3.1. Alakerran näytöt



Kuvio 23. Alakerran layout ja rampisto

Operointipaneelin sijoituspaikka huomioiden on loogista että valvomo-ohjelman päänäyttö on alakerran rullankäsittelylaitteiden ja rampiston yleisnäyttö. Siihen on yhdistetty rampistolle tulevien, siellä olevien sekä poistuvien rullien indikointi rulladatan bittitietojen ja laitteiden rajakytkintietojen mukaan. Rullien lisäksi on-line versiossa näytetään myös tulokuljettimella kulkevien rullien ramppiosoitetiedot sekä rullapinonnan viimeisen rullan data. Esitellyissä työpistesimulaation kuvakaappauksissa niitä ei näy, koska rulladatan siirto tapahtuu logiikan ohjelmassa eikä siten ollut käytettävissä työpisteen ollessa logiikkalaitteistosta irrallaan. Alakerran päänäyttöön päästään funktionäppäimellä F1.



Kuvio 24. Ramppien rajakytkintiedot

Mikäli käyttäjä haluaa tarkempaa diagnostiikkatietoa rampiston laitteiden toiminnasta, löytyy funktionäppäimen F2 painalluksella yksityiskohtaisempi rajakytkimien tilatietonäyttö. Tähän haluttiin koota taulukkomuodossa kaikkien ramppien kippaimien, annostelijoiden ja vastaanottimien rajakytkintietoihin perustuvat toimintatilat. Kippaimen toiminnasta ilmaistaan alaraja ja rulla tulee rampille tilat. Annostelijan osalta näytetään rulla on tieto sekä kaksisylinterisen vastaanottimen rajakombinaatiot: vastaanottoasento, keskiasento ja luovutusasento, jolloin rulla poistuu rampilta. Taulukon laitimmaisessa sarakkeessa on vielä vastaanottimen toimintatilat, joita ovat vastaanotto- ja luovutusasento. Taulukon vihreä väri indikoi rajatiedon on-tilaa ja harmaa puolestaan rajan ei-tilaa.



The image shows a control panel for a ramp system. At the top, there is a status bar with a logo, a date/time stamp (30.12.2010 14:15:26), and the title 'Rampit'. Below this, the panel is divided into two main sections for 'RAMPPI 3 (603)' and 'RAMPPI 4 (604)'. Each section contains a 'Datanäyttö' (Data display) field showing '0000000000000000', a 'Rullien lukumäärä' (Roll count) field showing '0', and a table of 'Rulla' (Roll) data. The table has two columns: 'Rulla on' (Roll is) and 'Viimeinen rulla' (Last roll), both with 'Arvo' (Value) fields. For Ramp 3, the values are all '0'. For Ramp 4, the values are also all '0'. To the right of each section are three buttons: 'EDITOINTI' (S11 for Ramp 3, S3 for Ramp 4), 'RESET / NOLLAUS' (S12 for Ramp 3, S4 for Ramp 4), and 'PAKKOPURKU' (S13 for Ramp 3, S5 for Ramp 4). The panel is surrounded by status indicators S1-S16 and function buttons F1-F20.

Kuvio 25. Data rampeilla

Yksi tärkeimpiä käytännön lisäarvoa tuovia tekijöitä operointipaneelin valvomo-ohjelmassa on mahdollisuus ramppikohtaisen datan esitykseen ja muokkaukseen. Rampiston 19 ramppia on tilansäästön vuoksi esitelty pareittain yhdeksässä näytössä ja tilaan päästään funktionäppäimellä F11. Aktiivisen rampin valinta tapahtuu funktionäppäimillä F10 (seuraava) ja F9 (edellinen). Kuvion 25. tilanäytöstä voidaan valita käyttäjää opastavin painikkein ramppikohtaisesti toiminnot: editointi, reset/nollaus ja pakkopurku. Näiden toimintojen valinta vie käyttäjän ramppidatan editointitilaan, jonka näyttö ja toiminnot on esitelty seuraavaksi.

**Rampit** 30.12.2010 14:16:01 Protacore

**RAMPPI 3 (603)**

Datasana: 0000000000000000

Rullien lukumäärä: 0

Edellinen ramppi F9  
Seuraava ramppi F10

**EDITOINTI TILA**

Paluu ilman muutosta (S9)  
Muuta Uudet Arvot (S10)  
RESET / NOLLAUS (S12)  
PAKKOPURKU (S13)

	Rulla on		Viimeinen rulla	
	Arvo	Uusi arvo	Arvo	Uusi arvo
Rulla 1	0	0	0	0
Rulla 2	0	0	0	0
Rulla 3	0	0	0	0
Rulla 4	0	0	0	0

**RAMPPI 4 (604)**

Datasana: 0000000000000000

Rullien lukumäärä: 0

	Rulla on	Viimeinen rulla
	Arvo	Arvo
Rulla 1	0	0
Rulla 2	0	0
Rulla 3	0	0
Rulla 4	0	0

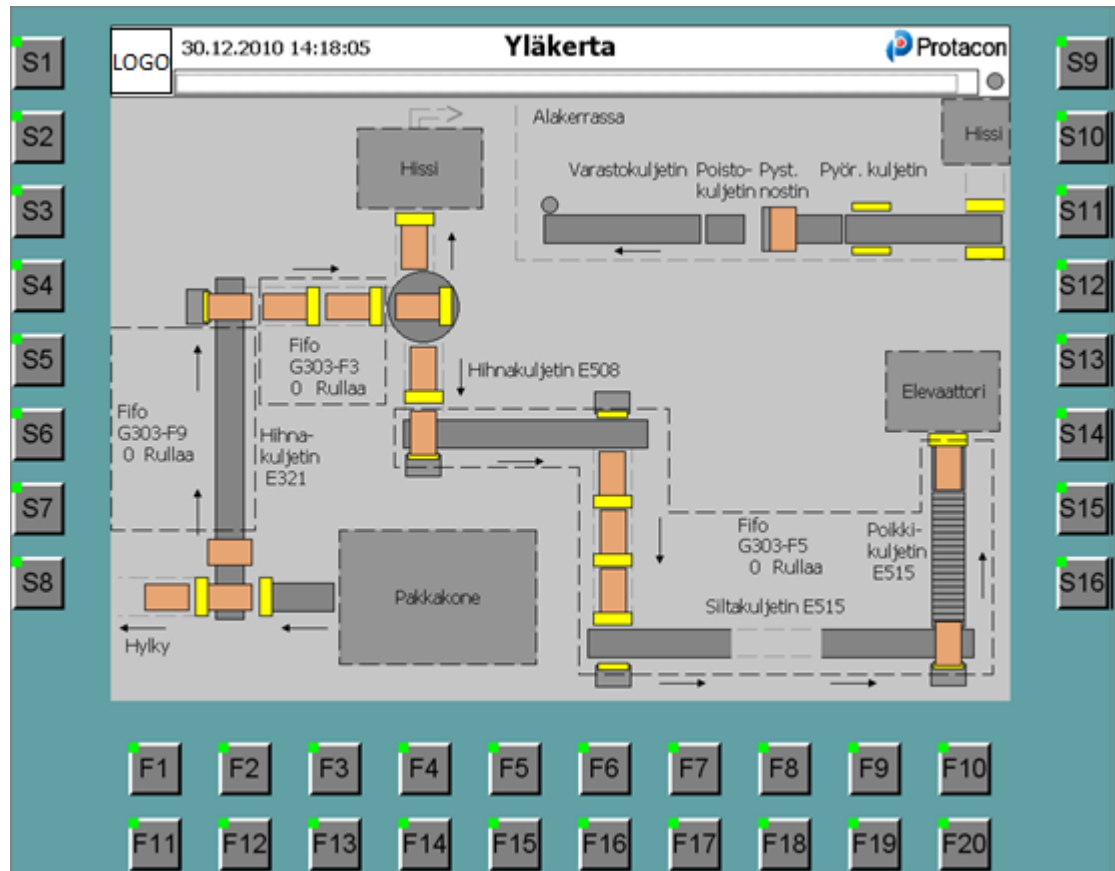
EDITOINTI (S3)  
RESET / NOLLAUS (S4)  
PAKKOPURKU (S5)

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10  
F11 F12 F13 F14 F15 F16 F17 F18 F19 F20

Kuvio 26. Ramppidatan editointi

Kun editointitila on valittu, alkaa aktiivisen rampin kohdalla vilkkumaan havainnollisuuden varmistamiseksi mustan ja punaisen väreissä teksti "EDITOINTI TILA." Mikäli toimintoon on tultu erehdyksessä, on editointitilan avautuvista toiminnoista ensimmäisenä painike "paluu ilman muutosta", jota painamalla palataan edelliseen tilaan tallentamatta mahdollisia muutoksia. Tilan päätarkoitus on kuitenkin avata ramppidatan editointitaulukko rullien purkua ja pinoamista ohjaavien "rulla on" ja "viimeinen rulla" bittien muuttamista varten. Taulukossa liikutaan kursorinäppäimin syöttökentästä toiseen ja arvo muutetaan paneelin numeronäppäimistöltä. Taulukon muutokset vahvistetaan "muuta uudet arvot" -painikkeella, jolloin uudet arvot tallennetaan ja palataan pois editointitilasta.

#### 4.3.2. Yläkerran näytöt



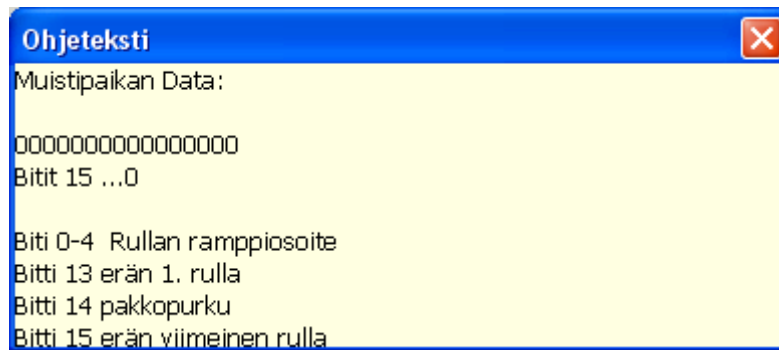
Kuvio 27. Yläkerran layout

Yläkerran layout näyttöön päästään funktionäppäimellä F3 ja samoin kuin alakerran vastaava, sen on tarkoitus toimia alueen tilatietoja kuvaavana yleisnäyttönä. Siinä näytetään yläkerran rullankäsittelylaitteiden lisäksi myös rullahissi ja muut alakerran kuljetinlaitteet arkkirullien käsittelyä varten. Rullien kulkua visualisoidaan rajatietojen lisäksi rulladatan kiinteiden paikkojen ja kuljettimien FIFO -rekisterien tiedoin. Logiikkaan kytkettynä näytetään on-line valvomonäytössä myös rullanpakkauskooneelta poistuvan rullan osoitedata.



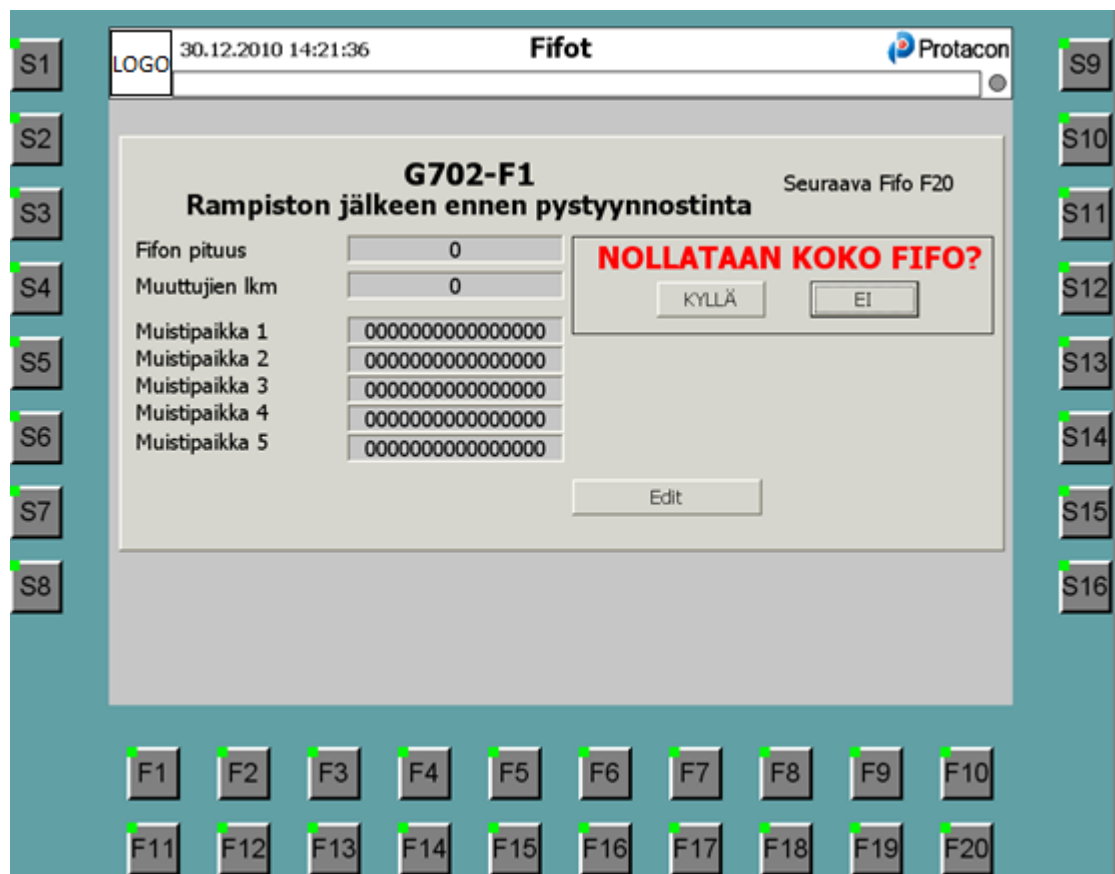
Kuvio 28. Kuljettimilla oleva rullamuisti: FIFO -rekisterit

Vastaavasti kuin ramppikohtaisen datan kohdalla, voidaan uudessa valvomo-ohjelmassa esittää ja muokata myös yksittäisiä FIFO -rekistereitä. Rekistereitä on neljä kappaletta: G702-F1, G303-F3, G303-F5 ja G303-F9, mutta toisin kuin ramppien tapauksessa, tässä ei nähty tarpeelliseksi niputtaa useampaa rekisteriä samaan näyttöön. Niputuksessa olisi vähäisen tilansäästön kustannuksella kärsinyt selkeys, sillä FIFO -rekisterit sisältävät runsaita määriä dataa. Tilaan päästään funktionäppäimellä F5 ja rekistereiden välillä liikutaan funktionäppäimin F20 (seuraava) ja F19 (edellinen). FIFO -rekisteristä esitetään sen pituuden lisäksi myös muuttujien lukumäärä sekä yksittäisten muistipaikkojen datasanat (16 bittiä 1/0 tietoina). Näytössä on myös rekisteriä ohjaavat painikkeet "Reset / Nollaus" ja "Edit", jotka johtavat kuvioiden 30 ja 31 näyttöihin.



Kuvio 29. FIFO -rekisterin Help-ikkuna

Kaikkien FIFO -rekisterien esitykseen ja muokkaukseen liittyvien näyttöjen yhteydessä on myös käytettävissä kuviossa 29 esitelty datasanan ohjeikkuna, joka saadaan esiin operointipaneelin HELP-näppäimellä.



Kuvio 30. Kuljettimilla oleva rullamuisti: FIFO -rekisterin resetointi

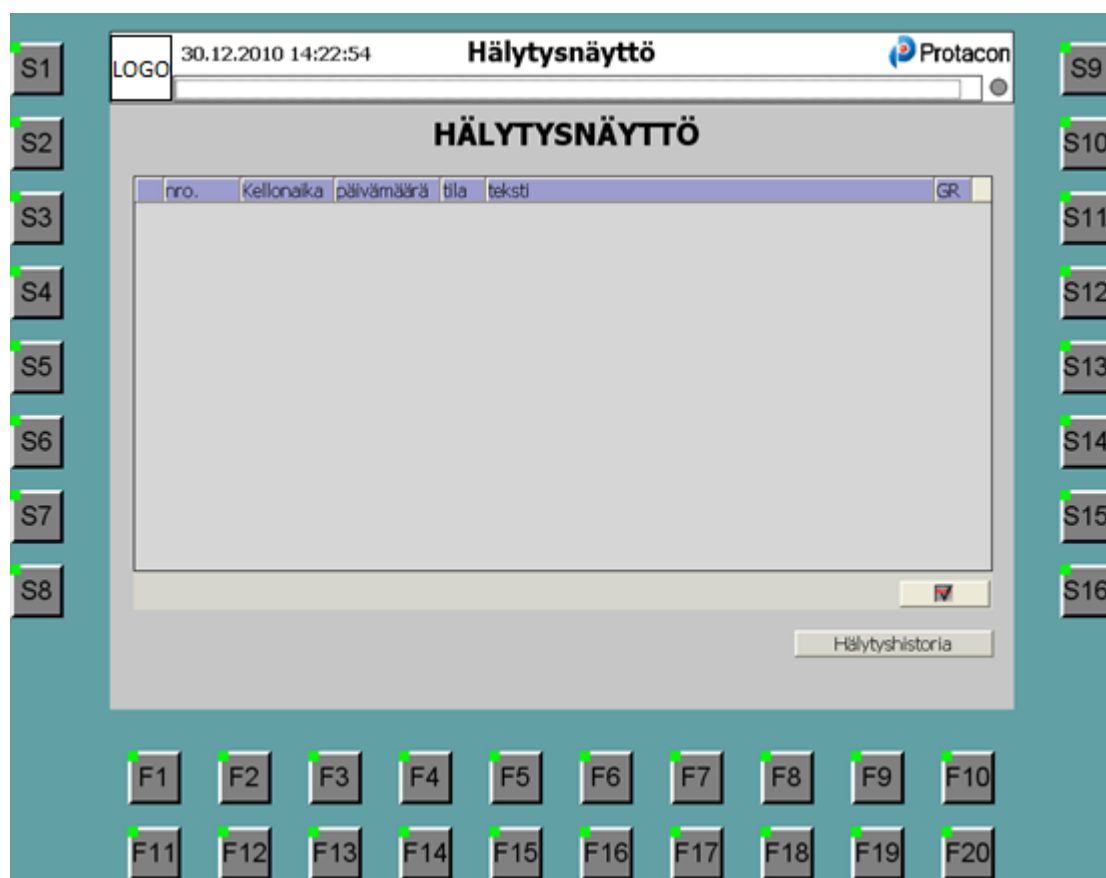
FIFO -rekisterin esitysnäytön "Reset / Nollaus" -painike aukaisee valintaikkunan, jossa toiminnon havainnollistamiseksi vielä vilkkuu mustan ja punaisen väreissä teksti

"NOLLATAANKO KOKO FIFO?". Käyttäjän valitessa "KYLLÄ", nollataan rekisterin data ja palataan edelliseen tilaan. Jos valitaan "EI", palataan edelliseen tilaan muutoksitta.

Kuvio 31. Kuljettimilla oleva rullamuisti: FIFO -rekisterin editointi

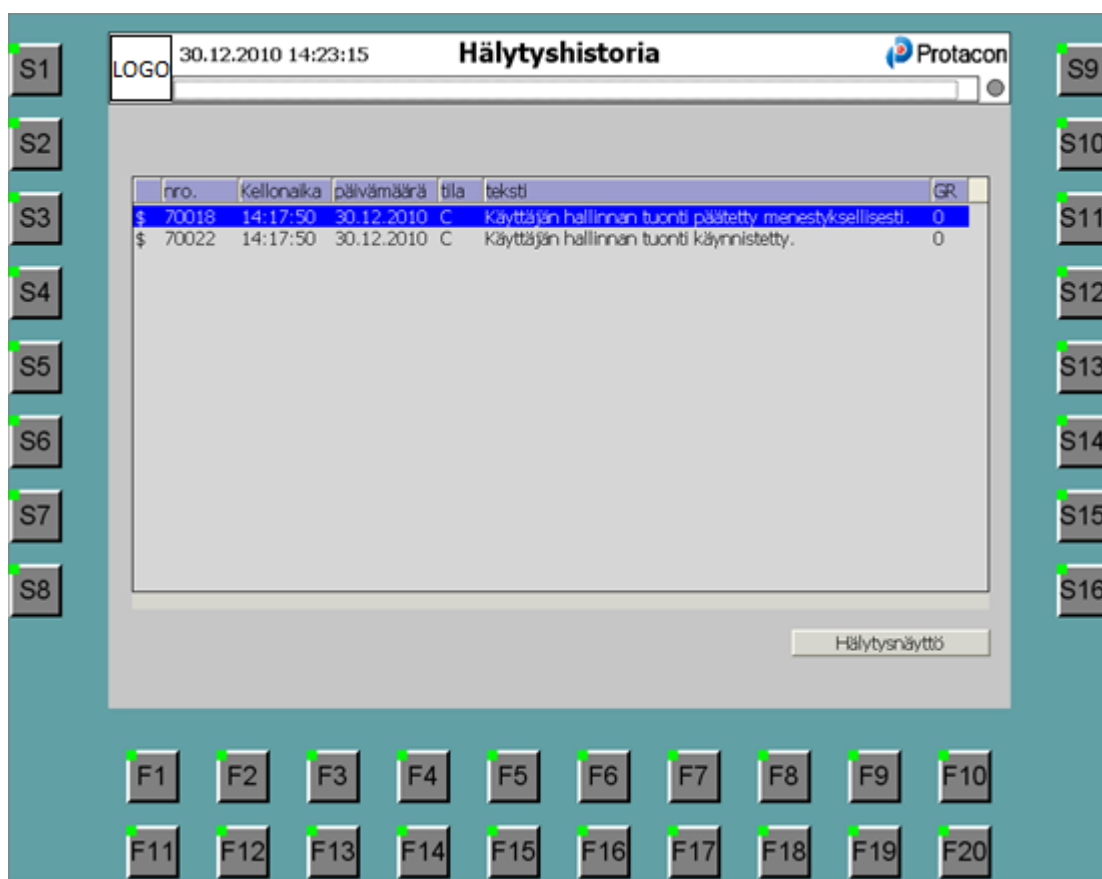
FIFO -rekisterin esitysnäytön "Edit" -painike tuo näytölle valintaikkunan, joka eroten edellisestä mahdollistaa rekisterin yksittäisten muistipaikkojen muokkauksen. Muokkauksella käytettäessä syötetään ensiksi "Muistipaikka nro" -kenttään käsiteltävän muistipaikan numero, jolloin valvomo-ohjelma hakee vastaavan muistipaikan datan esille. Seuraavaksi voidaan syöttää käsin "Muutettava data" -kenttään haluttu uusi bittijono ja painetaan "Muuta" tallennusta varten tai vaihtoehtoisesti voidaan poistaa kyseisen muistipaikan data "Poista" -painikkeella. "Paluu" -painike palaa edelliseen näyttöön tallentamatta muutoksia.

### 4.3.3. Hälytysnäytöt



Kuvio 32. Hälytysnäyttö

Operointipaneelin myötä järjestelmään saatiin myös tavanomaisesti käytössä olevaa pelkkää häiriövaloa huomattavasti havainnollisempi ja vianhaun kannalta käyttökelpoisempi työkalu pulmatilanteiden ratkomista varten. Funktionäppäimen F4 painallus tuo esille aktiivisten hälytysten näytön, jossa mahdolliset hälytykset listataan aikajärjestyksessä. Hälytykset liipaiseva ohjelma itsessään on osa logiikkaohjelmaa, ja operointipaneeli toimii tämän ohjelmakoodin näyttöpäätteenä. Hälytysnäytön ja seuraavaksi esiteltävän hälytyshistorianäytön välillä päästään liikkumaan joko funktionäppäimin tai vaihtoehtoisesti näytöllä sijaitsevin painikkein.



Kuvio 33. Hälytyshistoria

Käynnissä olevien hälytysten lisäksi uusi valvomo-ohjelma mahdollistaa myös jo kuitattujen sekä poistuneiden hälytysten tutkimisen. Hälytyshistoria saadaan esiin funktionäppäimellä F14 ja siihen tallennetaan ei-aktiiviset hälytykset aikaleimoin sekä värikoodein.

## 5 KOESTUS JA KÄYTTÖÖNOTTO

Logiikka- ja valvomo-ohjelmoinnin edettyä pisteeseen jossa yksittäisillä työasemilla tehtävien simulointien ohelle tarvittiin laitteistopohjaista testausta, rakennettiin toimitukseen hankituista logiikkalaitteista toimipaikan hallin puolelle koestusta varten FAT -ympäristö (Factory Acceptance Test, tehdastesti toimittajan tai valmistajan tiloissa). Modulaarisien logiikka- ja hajautusyksiköiden kokoaminen päätettiin tehdä kaikkien yksiköiden osalta kerralla valmiiksi. Tämän katsottiin helpottavan ja nopeuttavan paitsi testausta, myös toimitusta sekä asennusta tehtaalla. Sen sijaan että vierassa ympäristössä ryhdyttäisiin purkamaan kuhunkin yksikköön tarvittavia moduu-



leita laatikoista erilleen, niin nyt asentaja voi nostaa valmiit yksiköt suoraan laite-kaappeihin.



Kuvio 34. FAT -kokoontaminen

FAT -kokoontaminen muodostettiin tehdassijoittelua mukaillen. Tehtaalla ala- ja yläkerran sähkötiloihin sijoitetaan yhden 400-sarjan logiikan. Kuviossa 34 vasemmassa ylänurkassa näkyvä alakerran logiikka ei sisällä omia tuloja eikä lähtöjä, kun taas vastapäätä näkyvä yläkerran laite on kalustettu myös kenttäliityntöjen osalta I/O -moduulein sekä kuvassa vaaleina näkyvin S5-S7-adapterein. Logiikoiden jatkeena toimivien ET200M-yksiköiden toteutettava I/O-hajautus seuraa hyvin pitkälle korvautuvan järjestelmän rakennetta (kts. liitteet 2-4). Sen myötä alakerran PROFIBUS-väylään kytkeytyy 12 hajautusyksikköä ja yläkerran logiikan perässä hajautusyksiköitä on neljä. Yläkerran logiikan MPI -väylässä on lisäksi yksi vikatapauksia hoitava 300-sarjan erillislogiikka. Operointipaneeli kytkeytyy alakerran logiikan MPI -väylään.

Testaukseen oli varattu Jyväskylässä aikaa yksi viikko ja siihen osallistui sekä asiakkaan että Protaconin puolelta kaksi henkilöä. Testauksen aikana ladattiin sovellusohjelmat logiikoihin sekä operointipaneeliin ja niitä simuloitiin eri tilanteita mallintamalla ohjelmilta halutun toiminnallisuuden varmistamiseksi. Samalla varmistui myös logiikkalaitteiden toimivuus yhtenä kokonaisuutena, jolloin ne olivat valmiita tehtaalta toimitettaviksi. Testauksen aikana ohjelmaa korjattiin löytyneiden puutteiden osalta ja siihen tehtiin muutamia muutoksia asiakkaan toiveiden pohjalta.

Jyväskylässä tehdyn testauksen perusteella ei tietysti vielä pystytty näkemään tehtaalla ohjattavien laitteiden käyttäytymistä, sillä tulojen ja lähtöjen toiminta jäi simuloitunakin pelkästään tilaa ilmaisevien merkkivalojen muuttumiseksi logiikkalaitteissa. Testin perusteella kuitenkin voitiin todeta, ettei ohjelmassa ole mitään karkeita tai asennustilanteessa vaaraa aiheuttavia virheitä, joten tältä osin oli turvallista edetä käyttöönottovaiheeseen. Tehtaan päässä tämä tapahtuu kahdessa vaiheessa, vaikkakin peräkkäisinä viikkoina. Ensimmäisessä vaiheessa otetaan käyttöön rullanpakkaus-konetta edeltävät kuljettimet sekä vaippakäärintäasema. Toisessa vaiheessa on vuorossa rullanpakkaus-koneen jälkeiset kuljettimet sekä rullarampisto. Alustavaksi käyttöönottoajankohdaksi oli asiakkaan kanssa sovittu tammikuu 2011.

## 6 POHDINTA

Tämä opinnäytetyö jakautui sisällöllisesti kahteen osuuteen: osaltaan sovelluskehitys tapahtui kääntämällä ohjelmakoodia vanhoista ohjelmalistauksista sovittaen se uuteen ympäristöön ja toisaalta taas ohjelman toiminnallisuuteen kehitettiin myös uusia ominaisuuksia yhteistyössä asiakkaan kanssa. Näistä varsinkin käännöstyön määrä oli valtava. Harmikseni sain pian huomata työstäväni sitä suurimman osan ajasta yksikseni samalla kun kokeneemmat ohjelmoijat työskentelivät muiden käynnissä olevien projektien parissa. Toisinaan projektien viedessä heidät asiakkaiden kohteisiin, ei toimipisteeseen jäänyt ketään, jolta olisin voinut kysyä neuvoja pulmatilanteissa.

Tokihan pulmatilanteita matkalle myös mahtui, olihan projekti jo kokoluokaltaan jotain aivan muuta mihin koulun kursseilla tehdyt vastaavat ylsivät laajuudeltaan. Lisäksi kuljetinjärjestelmän ja rullarampiston toiminnan käsittäminen kokonaisuudessaan, saati sen hallitseminen itse koodatulla logiikkaohjelmalla tuntui projektin alkuvaiheessa kovin vaikealta. Työharjoittelujaksot paperitehtailla sekä automaatiopuolelle erikoistuvan koulutusohjelman myötä pääpiirteiltään tutuksi tulleet kuljetinjärjestelmät ja ohjelmointityökalut mahdollistivat kuitenkin työskentelyn kokeneen porukan osana. Vähitellen huomasinkin ymmärrykseni kasvavan ohjelmarakenteen pienien palasten kautta kohti suurempaa yhtenäisyyttä, ja alkujaan massiiviselta tuntunut vanha sovellusohjelma alkoi jäsentyä selkeämmiksi ohjelmanosasiksi. Toiset kut-

suvat asiaa ehkä ammattitaidoksi, toiset taas kokemukseksi, mutta omakohtaisesti oli mukava huomata itsensä kehittyneen projektin edetessä löytämään vanhasta ja niukahkosti kommentoidusta ohjelmakoodista yksittäisten laitteiden toimintaan vaikuttavia lukituksia, datavertailua ja niin edelleen.

Mitä kehitystyössä sitten olisi voinut ehkä tehdä toisin? Koska kyseessä oli kahden eroavan logiikkalaitteiston vuoksi eri sovellusohjelmaformaattit, olisi automaattisen käännöstyökalun puuttumisen vuoksi jo lähtökohtaiseksi työtavaksi voitu konversion sijaan valita alusta alkaen uuden sovelluksen kehitys. Jälkimmäiseen vaihtoehtoon olisi vaikuttanut vähemmän myös lähtötietojen heikkous sekä korvattavan järjestelmän termistön vieraus projektihenkilöstölle. Kun vielä paljastui asiakkaan toimittamien sovellusohjelmalistauksien vastaamattomuus tehtaan logiikkalaitteissa ajettaviin online-versioihin, lisääntyi työmäärä virtapiiristä virtapiiriin tehtävän tarkastuskierroksen ja sen mukaan tehtyjen muutoksien myötä vielä entisestään. Ei tietenkään pidä unohtaa etteikö sovelluskehitys tyhjältä pöydältä olisi myös aikaa vievää ja haastavaa työtä, mutta kaikkien nyt esiintyneiden ongelmien kautta se olisi saattanut olla kuitenkin jopa kannattavampi vaihtoehto. Ohjelmakoodista olisi samalla jalostunut pois myös menneiden vuosikymmenien vähätehoisemmista logiikkalaitteista parhaimmat tehot optimoivat, joskin vaikeaselkoiset ohjelmointitavat. Elinkaariajattelun kannalta tämä olisi helpottanut niin käyttöönottoa, monitorointia, kunnossapittoa kuin tulevaisuuden tuotekehitystäkin.

Työn tuloksellisuuden kannalta saatiin arvokasta tietoa FAT -vaiheessa, jonka perusteella voi sanoa tämän opinnäytetyön täyttävän sille asetetut tavoitteet hyvin. Logiikkaohjelmat sovitettiin uudelle laitteistoalustalle ja järjestelmään lisättiin toivottu operointipaneeli kuljettimien ja rullarampiston tilatietoja havainnollistamaan. Henkilökohtaisesti koen kehittyneeni projektityöskentelijänä, mutta vielä suuremmat harppaukset osaamisessani keskittyvät tietenkin ohjelmoitavien logiikoiden sovellussuunnitteluun ja laitteistokokonaisuuksien hahmottamiseen. Yhden projektin perusteella en varmasti pidä käsissäni vielä kaikkia alan avaimia, mutta näen opinnäytetyön aikana kartutetun tietotaidon lisäävän omia kykyjäni logiikkaohjelmien suunnittelijana.

## LÄHTEET

Protacon Group. 2011. Yrityksen kotisivut. Viitattu 9.3.2011.  
<http://www.protacon.fi/yritys>

KnowPap. 2010. Paperitekniikan ja tehtaan automaation oppimisympäristö. Viitattu 15.12.2010. M:\OPI\Teknologia\Opetusmateriaali\KnowPap\

Leminen, J. 2010. Lead engineer, Protacon Oy. Haastattelu 13.12.2010.

Parr, E.A. 1999. Programmable Controllers, An engineer's guide. Oxford: Newnes.

Fonselius, J., Pekkola, K., Selosmaa, S., Ström, M. & Välimaa, T. 1996. Koneautomaatio, AUTOMAATIOALAITTEET. Helsinki: Oy Edita Ab.

Siemens AG. 2011. Yrityksen tukisivut, S7-300 module data. Viitattu 9.3.2011.  
[http://support.automation.siemens.com/WW/llisapi.dll/csfetch/8859629/s7300\\_module\\_data\\_manual\\_en-US\\_en-US.pdf?func=cslib.csFetch&nodeid=10072094&force\\_download=true](http://support.automation.siemens.com/WW/llisapi.dll/csfetch/8859629/s7300_module_data_manual_en-US_en-US.pdf?func=cslib.csFetch&nodeid=10072094&force_download=true)

Siemens AG. 2011. Yrityksen tukisivut, S7-300 CPU 31xC and CPU 31x: Technical specifications. Viitattu 9.3.2011. [http://support.automation.siemens.com/WW/llisapi.dll/csfetch/12996906/s7300\\_cpu\\_31xc\\_and\\_cpu\\_31x\\_manual\\_en-US\\_en-US.pdf?func=cslib.csFetch&nodeid=13150809&forcedownload=true](http://support.automation.siemens.com/WW/llisapi.dll/csfetch/12996906/s7300_cpu_31xc_and_cpu_31x_manual_en-US_en-US.pdf?func=cslib.csFetch&nodeid=13150809&forcedownload=true)

SESKO Ry. 2011. Suomen sähkö- ja elektroniikka-alan standardisoinnista. Viitattu 9.3.2011. [http://www.sesko.fi/portal/fi/ajankohtaista/uudet\\_julkaisut?bid=395](http://www.sesko.fi/portal/fi/ajankohtaista/uudet_julkaisut?bid=395)

Siemens AG. 2011. Yrityksen suomalaiset kotisivut, Step 7 esittely. Viitattu 9.3.2011.  
[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat\\_logiikat\\_simatic/ohjelmistot/classic.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic/ohjelmistot/classic.htm)

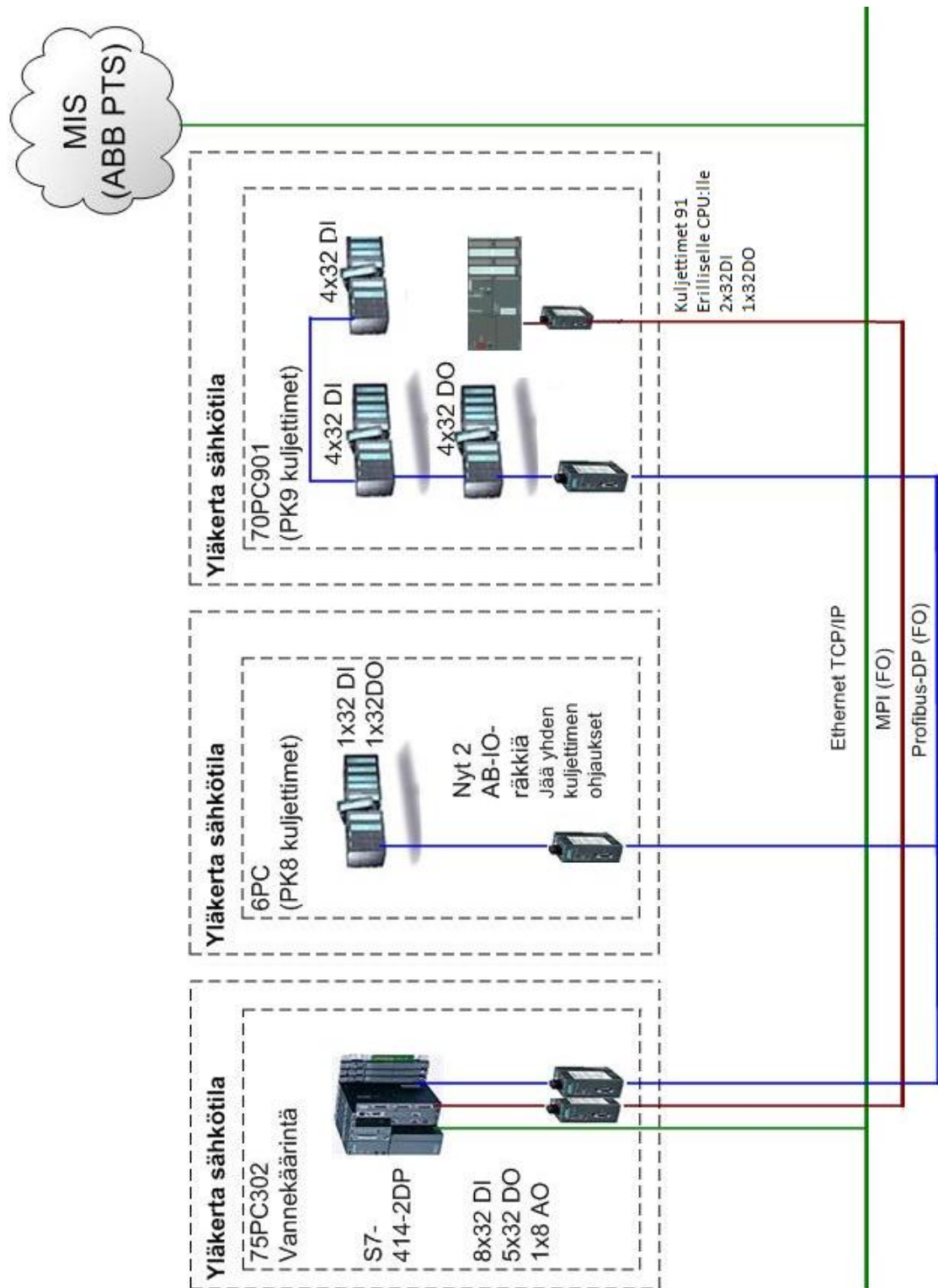
Siemens AG. 2011. Yrityksen suomalaiset kotisivut, WinCC esittely. Viitattu 9.3.2011.  
[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/automaatiotekniikka/kayttoliittymat/ohjelmistot/paneelien\\_ohjelmointi\\_wincc\\_flexible.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/kayttoliittymat/ohjelmistot/paneelien_ohjelmointi_wincc_flexible.htm)

Siemens AG. 2011. Yrityksen tukisivut, MP377 12" Keypad. Viitattu 9.3.2011.  
<http://support.automation.siemens.com/WW/llisapi.dll/30363188?func=ll&objId=30363188&objAction=csView&nodeid=10805570&lang=en&siteid=cseus&aktprim=0&extranet=standard&viewreg=WW&load=content>





Liite 3. Logiikkalaitteisto, pakkaamon etupuoliset kuljettimet



Liite 4. Logiikkalaitteisto, pakkaamon jälkeiset kuljettimet

